ICAS 2020 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

AN ONTOLOGICAL APPROACH FOR CLASSIFYING AIRCRAFT ACTUATORS

Felix Larsson[1,2], Christopher Reichenwallner[2], Alessandro Dell'Amico[1,2] & Petter Krus[2]

[1]Saab AB, SE-581 88, Linköping, Sweden
[2]Linköping University, SE-581 83, Linköping, Sweden

Abstract

Today, electrification is a widely discussed topic within all fields of engineering and especially in the field of aeronautics. Whether electrification of aircraft systems is the only viable option to overcome environmental issues is a highly complex question which involves many aspects. In this paper, fundamental differences between electrified and hydraulic aircraft actuators are mapped, compared and discussed to understand how and if these systems can be compared without misleading results.

In this paper, different actuator architectures were collected from literature, classified and analyzed by using an ontological approach. The classification was created by utilizing defined classes where the class definition corresponded to the functional requirements which had to be fulfilled by the actuator architecture.

This work has shown that ontologies can sufficiently be used for classification of actuator architectures. The built in reasoner can be used to sort a large set of actuators into comparable classes. Meanwhile, the ontology defines a framework where important information of the actuators and their components can be stored and structured.

Keywords: Aircraft actuators, EMA, EHA, SHA, Ontology

1. Introduction

Today, electrification is a widely discussed topic within all fields of engineering, especially in the field of aeronautics. Whether electrification of aircraft systems is the preferred future alternative is a highly complex question with many aspects involved.

The flight control actuation system is one of the systems currently under investigation for electrification. The aircraft actuation system can be divided into two different groups, primary and secondary flight actuation system. The primary actuation system is a flight critical system which controls the aircraft's yaw, pitch and roll maneuvers. A single failure of any component in the aircraft cannot lead to a loss of function of a flight critical system since the aircraft's operation is dependent on it. Reliability is therefore one of the primary flight actuation system's most important requirements. Enough reliability can either be achieved by designing a system which will never fail, which is not easily achieved, or by creating tolerance to failure by adding redundancy.[1]

Currently, there are three fundamentally different aircraft actuator types in focus, the servo-hydraulic actuator (SHA), the electro-hydrostatic actuator (EHA) and the electromechanical actuator (EMA). [2], [3], [4] The SHA is the conventional alternative with many years of in-service experience, high power density and high reliability. The EHA and EMA are seen as the future alternatives for electrification and are more commonly equipped as back-up actuators, or as secondary flight actuators.[5], [6]. The primary drivers for electrification are the possibility of power on demand, energy management, ease of maintenance and increased efficiency. The EHA is currently used for primary flight control in the F-35 fighter aircraft, while the EMA is more commonly used for back-up or secondary flight control. One of the reasons why EMAs are difficult to use for primary flight control is the risk of jamming. This risk is currently investigated within research and possible solutions for mitigating it have been found, but there are not yet many jam-free devices being produced [7] [8] [9].

A schematic overview of the differences between these technologies is presented in figure 1, where examples of both simplex and dual-redundant actuators are presented.

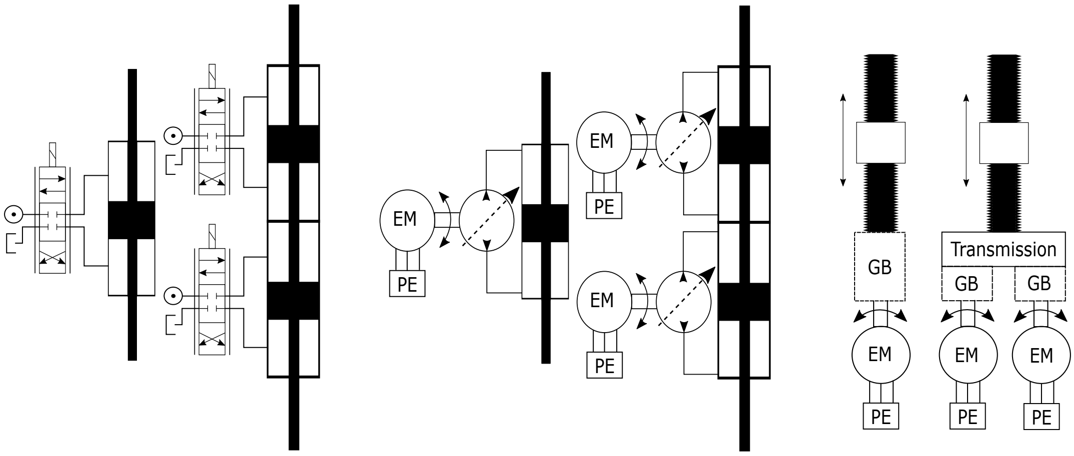


Figure 1 – Overview of different SHA, EHA and EMA architectures. EM, PE and GB stands for Electric Motor, Power Electronics and Gear Box respectively.

The components utilized to create the fundamental actuation functions with SHAs are a centralized hydraulic supply system, a servo valve and a hydraulic cylinder. To make this architecture dual-redundant, two supply systems connected to separated servo valves are often used. Then, either one tandem (dual chamber) cylinder, or two separated parallel cylinders can be used to create the linear mechanical power from the metered hydraulic power outputted from the servo valves.

Parallel or tandem cylinders are also utilized in EHAs, but the hydraulic power supply and control are created locally with electric motors connected to hydraulic pumps. Here, the pumps can either have variable displacement, which make it possible to control the hydraulic power with the pumps, or fixed displacement which require power control through the input rotational power provided by the electric motors. The electric motors also require separated power electronics (one unit per motor or channel) to supply and control them. During the literature review, only architectures using one or several separated pump units (electric motor+pump) were found.

The single channel EMA most often consist of a single electric motor connected either to a gearbox, or directly to a ball screw, where the ball screw is used to translate the rotational power into linear. Power electronics is necessary for the electric motor in EMAs as well. The found dual-channel architectures most often only utilize one single screw mechanism, while the electric motors and gearboxes are duplex. Either two separated electric motors, or one dual channel electric motor is used. If two separated electric motors are used, some transmission to sum either the rotational velocity, or torque produced by the motors is necessary. There are also examples where dual-redundant screw mechanisms are utilized to create full duplex actuators.

Today, it is difficult to determine which technology is the best. The optimal solution may highly depend on the intended platform, since various platforms will set different requirements on the actuators. These requirements can be divided into two groups; functional and performance requirements. Examples of these can be found in table 1. When comparing the different technologies, it is important to understand how the requirements will affect the attributes to be compared. In other words, it is important to ensure comparability of the actuators to be compared, which can be done by ensuring they are fulfilling the same set of requirements. Otherwise, the comparison might lead to misleading results.

Table 1 – Examples of functional- and performance requirements.

Functional Requirement	**Performance Requirement**
Control rudder position	Response
Damped mode	Force
Back-drivable	Bandwidth
Capable of de-clutching	No-load velocity
Full performance in case of failure	Stiffness

The performance requirements set on the actuators by a specific platform can most often be fulfilled by sizing and design of specific components performing the primary actuating function. Thereby, to certain limit, all actuator technologies can be used sufficiently. However, to ensure fulfillment of the functional requirements, supplementary components or functions might have to be added to the system. These additions will be different for SHAs, EMAs and EHAs, and will thereby highly influence the outcomes from a comparison.

In this paper, a method for ensuring comparability of different actuator concepts by proving fulfillment of functional requirements is proposed. The research question handled is: How can comparability between different actuator technologies be ensured before performance aspects are included?

2. Functional modeling

This section presents the proposed methodology for functional modeling and classification of different actuator architectures. A schematic view of the methodology is presented in figure 2. The proposed method includes an ontological approach, where the language "OWL" is utilized. This can be realized by using the open-source editor "Protegé". More information regarding ontologies, its use, OWL and Protegé can be read in these articles: In [10], the fundamentals of ontology modeling using Protegé and OWL can be read. More detailed knowledge of ontology modeling and how to use for example relation chains can be read in [11]. Applications of ontological modeling within engineering domains can be read here [12] [13] [14] [15] [16].

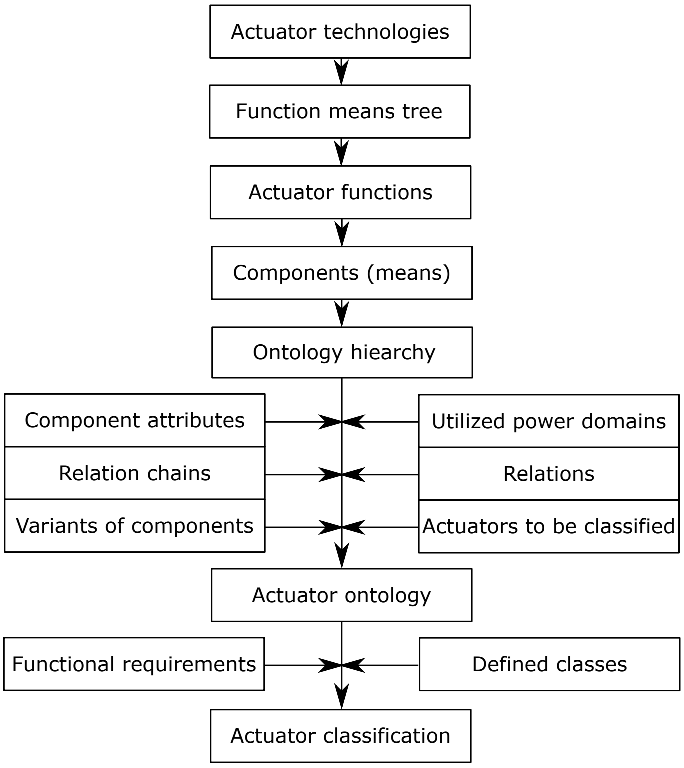


Figure 2 – Schematic view of the proposed methodology.

2.1 Actuator functions and means

To map up the fundamental functions an actuator must perform, a function means tree can be created. If an actuator is seen as a black box, it will receive certain power input (Create power), which it then has to convert (Convert power) and control (Create Power control).

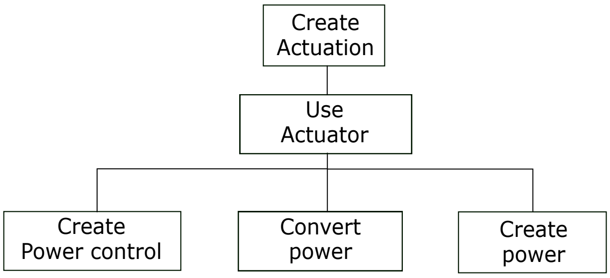


Figure 3 – A function-means tree of an unspecified actuator.

Commonly used means, or components, to perform the actuation functions presented in figure 3 are specified in table 2. To further understand how the actuator technologies (SHA, EHA and EMA) differentiate from each other in internal functions, these functions can be further detailed into technology-specific functions. Since this work only will focus on linear primary flight control actuators, this will

Table 2 – Functions and means.

Create actuation	**Convert power**
Use linear position controlled actuator	Use rectifier
Use rotational position controlled actuator	Use electric motor
Use force controlled actuator	Use servo valve
Use speed controlled actuator	Use screw
Create power	Use inverter
Use hydraulic supply	Use hydraulic pump
Use electric supply	Use gearbox
Create power control	Use hydraulic motor
Use VD-motor	Use hydraulic cylinder
Use power electronics	-
Use servo valve	-
Use VD-pump	-

only be done for such architectures. A flow-chart of power domains and functions is presented in figure 4.

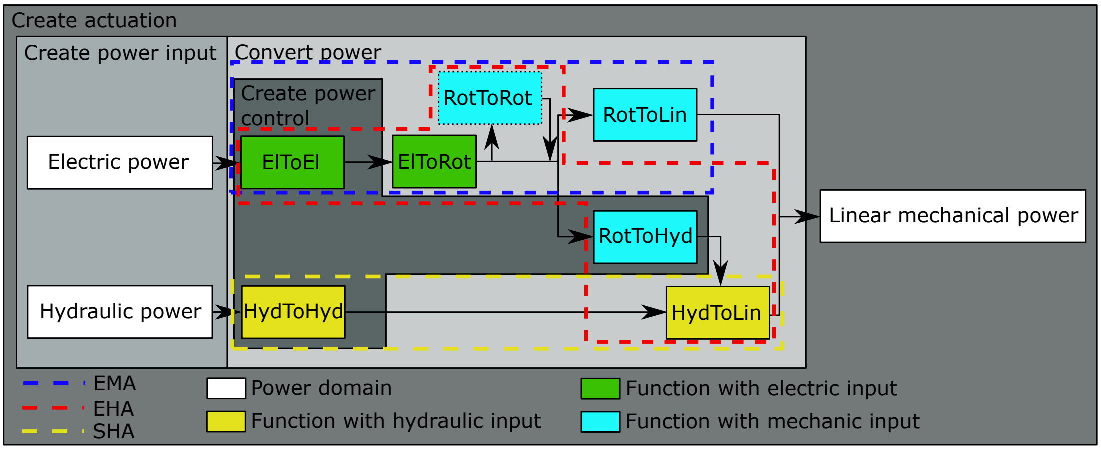


Figure 4 – A flow chart of the internal actuator functions and power domains.

For an electrically supplied actuator, the power control often takes place within the power electronics, where the electric power also is transformed (ElToEl). Then the transformed controlled electric power is converted into rotational mechanical power (ElToRot). After this, the rotational mechanical power is either transformed (RotToRot), directly converted into linear mechanical power (RotToLin) or hydraulic power (RotToHyd) depending on if the actuator is an EMA or EHA. Within the function RotToHyd, the hydraulic power output may also, in some cases, be controlled depending on what mean is used. If the rotational mechanical power is converted into hydraulic power, it also has to be converted into linear mechanical power (HydToLin). For an hydraulically supplied actuator (SHA in this case), the distributed hydraulic power is firstly metered, i.e. transformed and controlled (HydToHyd), and then converted directly into linear mechanical power (HydToLin).

The components presented in table 2 can now be assigned as means to the more detailed functions, as shown in table 3.

2.2 Ontology hierarchy

To model different actuators by using an ontology, one approach is to connect the involving components to each other and let them together form the actuator. By then describing the functions and attributes of the components, the ontology will hold necessary information about the modeled actuators to classify them. To do this, a hierarchy of classes must be created.

Table 3 – More detailed power conversion functions and their means.

Convert power		
Electric input	**Rotational mechanic input**	**Hydraulic input**
ElToEl	**RotToRot**	**HydToHyd**
Use rectifier	Use gearbox	Use servo valve
Use inverter	-	-
ElToRot	**RotToLin**	**HydToRot**
Use electric motor	Use screw	Use hydraulic motor
-	**RotToHyd**	**HydToLin**
-	Use hydraulic pump	Use hydraulic cylinder

The functions and means mapped in the previous section are used as a foundation for the ontology and will thereby be modeled as top level classes. The class called "Actuator" will include all actuator architectures to be modeled, "Function" will include all functions, "Mean" will include all actuator components and "Power Domain" will include all different power domains the functions will utilize. Within these classes, appropriate sub-classes are created to structure all different entities. For example, an appropriate sub-class under "Mean" can be "HydraulicMean". This is then repeated until the lowest level in the hierarchy is created. An example is shown in figure 5.

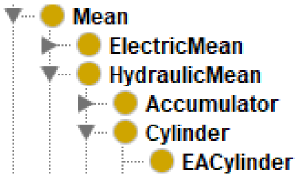


Figure 5 – This figure shows an example of an appropriate hierarchy for the ontology. EACylinder stands for Equal Area cylinder.

When the hierarchy is completed, individuals to each lowest level sub-class must be added. These will later represent for example the components to be assigned to the actuator architectures.

After this, the ontology shall contain classes and individuals of all actuators, functions, components and power domains for linear actuators which was defined in the previous section. However, no relationships between these entities are defined. This means that there is not yet any information within the ontology which expresses how the different entities relate to each other. Relationships can beneficially be added to the classes to make sure that the ontology posses all information, but in this project, the focus was on the individuals since they will be used for both the classification and the composition of actuators. Thereby, the hierarchy of classes will be used for organizational purposes rather than for the classification, and relationships between them will therefore not be prioritized.

2.3 Actuator ontology

Within this section, the ontological modeling is presented. Here the hierarchy which was previously defined will be further developed with a higher level of detail and with internal relationships between the entities. When this is completed, the ontology will be fully fledged and thereby the classification part of the work can be included.

2.3.1 Actuators to be classified

In the top-level within the ontology hierarchy, a class named Actuators was added. This class will contain all actuator architectures which are to be classified. An actuator architecture is here defined as the composition of components which together builds the actuator. In this work, a literature review was conducted to find as many different actuator architectures as possible to add to the classification. However, the functionality of the ontology will be the same regardless of how many actuators are added. Some of the found actuator architectures are presented 4 and will be used for the classification.

Table 4 – Actuator architectures. The trim actuator included in the Saab 2000 SHA is used to position the wing to a neutral position in case of double failures.

Actuator name	Motor type	Pump	Linkage type	Valves	Gearbox
EPAD EHA [4]	PMSM	Bi-directional bent-axis pump	Balanced cylinder	SOV, PRV, check valves	Direct drive
J/IST/Moog EHA [17]	4xBLDC	4xFixed displacement pump	Single cylinder	SOV, PRV, ACV	Direct drive
EPAD EMA [18]	2xBLDC with damped mode functionality	-	Single ball screw	-	Velocity summing differential
EMAS EMA [19]	2xBLDC	-	Single ball screw	-	One Gearbox per motor
Saab 2000 SHA with trim actuator [20] [21]	-	-	2xBalanced cylinders	2xSOV 2xPRV, 2xACV, 2xFlapper nozzle servo valves	-
Parker horizontal stabilizer trim actuator [22]	2xBLDC	-	Duplex ball screw with brake	-	Torque summing gear train

2.3.2 Variants of components

If the actuators to be classified contain other components than the ones which were mapped, these must be added to the ontology in the same way as the other components. It is also important to have in mind that components might have been designed with different attributes, even though they have the same name. For example, a ball screw can be designed to be reversible, or not, and this makes two variants of the same component in the ontology, which should be modeled as two individuals, but they may be located in the same class. To summarize, this reasoning would result in one class (ball screw) with two individuals (reversible ball screw and non-reversible ball screw). From the actuators presented in table 4, the component variants presented in table 5 were found.

Table 5 – Variants of components extracted from table 4.

Electric motors	Hydraulic pumps	Linkages	Valves	Gearboxes
BLDC	Bi-directional bent-axis pump	Ball screw	Flapper nozzle	Velocity summing differential
PMSM	Fixed displacement pump	Duplex ball screw with brake	PRV, SOV, ACV, check valves	Torque summing gear train
BLDC with damped mode functionality	-	Balanced cylinder	-	Gearbox
-	-	Trim actuator	-	-

2.3.3 Component attributes

To further describe the component variants, attributes are added to the ontology. Here, the more information which can be used to describe a certain component, the better. Similar to the previous example with reversible and non-reversible ball screws, one PMSM (Permanent Magnet Synchronous Motor) might be designed to be dual-redundant through the equipment of dual channels, while another might only have one channel. In these two examples, two individuals of each component shall be created, and these shall in turn be described with different attributes. Appropriate attributes for these examples may be "reversible" and "dual redundant" respectively. Another example is a hydraulic cylinder which can have equal areas of its both chambers, which thereby implies that it has the attribute of equal flow to and from each chamber. All such attributes must be mapped for each component to be utilized within the ontology, especially those which will be used to classify the actuators. When mapped, they are added to the ontology in similar manner as previously.

2.3.4 Relations

When all individuals are added to the ontology, the relations between the entities which together form an ontological description of the actuator architectures must be established. The relations are modeled as object properties within Protegé. An example of a relation is "hasComponent", which is used to connect a component to an actuator. If an actuator only use one component to create a certain function, then the relation "hasComponent" can be used, but if it use several components for the same function in order to create redundancy, two different object properties are required in order to differentiate the relations. This since it is not be possible to connect the same component to the same actuator twice with the same relationship.

The relations which was used within this work are presented in table 6. Most of these relations will also have inverse relations. For example, if an actuator is described with "hasComponent" "BLDC", then the ontology should also understand that "BLDC" "isComponentOf" the certain actuator it is connected to. Therefore, appropriate inverse relations shall also be modeled as object properties.

Table 6 – Object properties (Relations) used in this work and their purposes.

Object property (Relation)	**Purpose**
hasComponent hasComponent1 hasComponent2 hasComponent3 hasComponent4	Connect components (means) to actuators
hasFunction	Connect components to functions
hasPowerSource	Connect actuators to power sources
Input Output	Connect input and output power domains to functions
hasAttribute	Connect components to attributes
hasComponentWithFunction hasComponentWithFunction1 hasComponentWithFunction2 hasComponentWithFunction3 hasComponentWithFunction4 hasComponentWithAttribute utilizesPowerDomain	Connect component functions, attributes and power domains to actuators (Relation chains)

2.3.5 Relation chains

Some relations can automatically be inferred by the ontology. If it is known that a chain of relations imply a new relationship, for example if an actuator has a component, which in turn has a function, it can be said that the actuator has a relationship to the components function directly. Such inferred relationships are necessary for the final classification, since the ontology will not understand how the actuator itself is connected to the functions of the components otherwise.

Figure 6 show an example of how an actuator now is modeled within the ontology. In the figure, the inferred relation chains can also be seen. By only connecting the actuator "EMA1" to the component "Dual-coil EM" with the relation "hasComponent", it inherit many relationships since the component itself has an attribute and a function, where the function also input and output different power domains. Here, the purple circles represents individuals, the big circles represents classes and the arrows are object properties which show the relationships between the individuals.

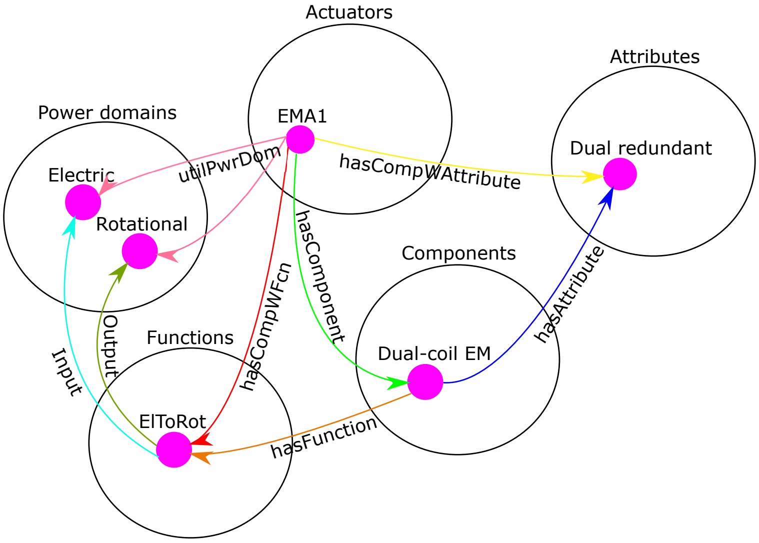


Figure 6 – A schematic sketch of a modeled actuator within the ontology.

2.4 Actuator classification

The actuator classification was implemented by creating defined classes in which accepted actuators are inferred by Protegé's built in reasoner. An actuator is accepted if it fulfills the requirements set in the defined class. For example, one defined class can be "EMA", in which EMAs can belong. Then, the requirements of the defined class "EMA" may be: actuators which utilize components with the functions "electric power control", "electric to rotational power conversion" and "rotational to linear power conversion".

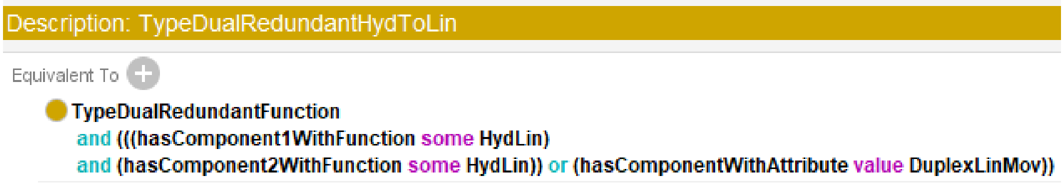
If an actuator belong to a certain class, it will be known to the ontology. This simplifies further classification since a new defined class can use previously defined classes, such as "EMA", as requirements for inclusion instead of expressing all the requirements which where set by the defined class "EMA".

To exemplify this, the functional requirement described in table 7 is used. This requirement can be applied for each function a certain actuation technology require in order to perform the actuating function, and will thereby create several functional requirements. These are realized by creating defined classes for each functional requirement.

Table 7 – Description of the functional requirement called "Dual-redundant function".

Functional Requirement	**Description**	**Applies to**
Dual-redundant function	Be able to maintain functionality if one channel performing the specific function is lost	All functions required by actuator to perform rudder control

One function linear hydraulic actuators must perform is "convert hydraulic power into linear mechanic power". The defined class related to this function may be called "TypeDualRedundantHydToLin" and is defined to include all actuators which either has two components performing the function, or one component which is dual-redundant. The realization of this is presented in figure 7.



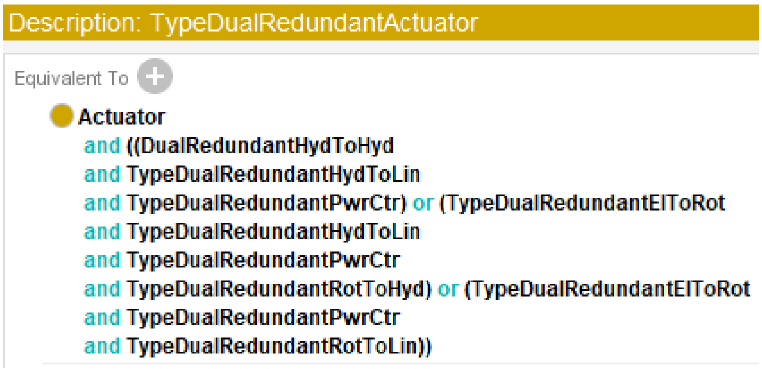
Description: TypeDualRedundantHydToLin

Equivalent To

```
TypeDualRedundantFunction
 and (((hasComponent1WithFunction some HydLin)
 and (hasComponent2WithFunction some HydLin)) or (hasComponentWithAttribute value DuplexLinMov))
```

Figure 7 – A defined class which will include all actuator concepts which has a dual redundant solution for the function "HydToLin".

When all functional requirements are created as defined classes, the actuators which fulfill these will be classified into these by the built in reasoner. Then, the class "TypeDualRedundantActuator" can be created. The intention of this class is to group all actuators which functions are performed only with dual-redundant components. Thereby, an actuator which for example utilize a dual-redundant electric motor, but not a dual-redundant ball screw, will not belong to this class. In figure 4 all fundamental functions performed by the different actuator technologies (SHA, EHA and EMA) are shown. The "TypeDualRedundantActuator" class is then defined as figure 8 shows.



Description: TypeDualRedundantActuator

Equivalent To

```
Actuator
 and ((DualRedundantHydToHyd
 and TypeDualRedundantHydToLin
 and TypeDualRedundantPwrCtr) or (TypeDualRedundantElToRot
 and TypeDualRedundantHydToLin
 and TypeDualRedundantPwrCtr
 and TypeDualRedundantRotToHyd) or (TypeDualRedundantElToRot
 and TypeDualRedundantPwrCtr
 and TypeDualRedundantRotToLin))
```

Figure 8 – Show the requirements which must be fulfilled of an actuator to belong to the class "TypeDualRedundantActuator".

2.4.1 Modes of operation

When investigating the comparability of actuators and the focus is on dual-redundant actuators, modes of operation during failure of one channel is of great importance. If one actuator architecture is designed to maintain full performance with one channel, its function during failure is different to an architecture where the performance is decreased in case of failure.

If a fail occur in an dual-channel actuator the outcome can be no loss of function, temporary loss of function or loss of function, depending on what or where the error occur. If the result is no loss of function, the actuator is "fail-operative" and maintain full performance, or the performance is reduced and the actuator is then said to be "fail-functional". If the failure cause a temporary loss of function, the actuator may be reconfigured and thereby either achieve an "fail-operative" or "fail-functional" mode of operation. If the function can not be maintained, the actuator will enter the "fail-safe" mode where its response to failure may differ depending on the design or the fault. The actuator may either lock in the position where it failed, float freely with the wing without disturbing other systems, or move to a predetermined position where it do not disturb other systems. These states are called "Fail-freeze", "Fail-passive" or "Fail-neutral" respectively. [1]

The main reason for usage of redundancy is to have the ability to remain fail-operative or fail-functional. If one channel fail, then the other can maintain performance of the system. However, if two actuators are dual redundant and will maintain performance in case of a single failure, but their response to a second failure differ, they may not be classified into being comparable since they are designed to perform different functions.

In similar manner as previously, three new defined classes are created to group the actuator architectures with respect to their response to failure. With regards to the functions and attributes of the components used in the actuator architectures, it can be determined if damped mode (Fail-passive), locking the actuator output in position (Fail-freeze), or returning the actuator position into a neutral position (Fail-neutral) is achievable. Then, three sub-classes to the class "TypeDualRedundantActuator" can be created by use of the defined classes previously described. The purpose of these sub-classes is to further classify the dual redundant actuators with respect to their measure to failure, namely "Fail-freeze", "Fail-passive" or "Fail-neutral". Thereby, dual redundant actuators with the same response to failure can be found.

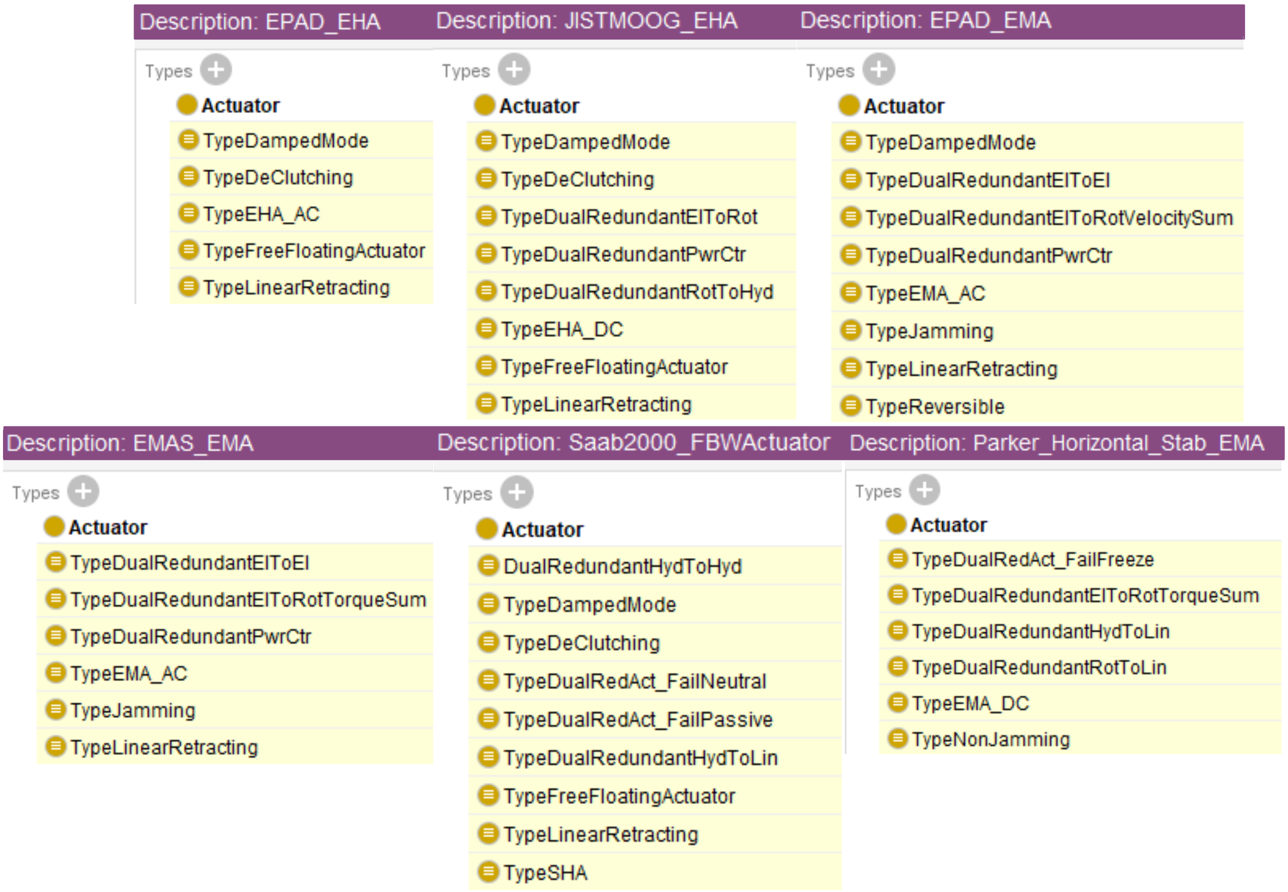
Other classes which are included in the ontology, but currently not used for the classification are: type of actuator (SHA, EMA or EHA), type of power supply (electric AC, electric DC or hydraulic), movement output principle (retracting or non-retracting linear movement), capability of jamming, capability of freely floating and how electric motor power is added when dual channels are equipped (torque- or velocity summing).

3. Results from the classification

In this section, the results of the created ontology is presented. The focus is to see in which defined classes the actuator architectures presented in table 4 belong according to the built in reasoner in Protegé.

In figure 9 the descriptions of the different actuators are presented. It can be seen which sub-classes they are inferred to belong in (the yellow-marked defined classes present under the class "Actuator"). According to the reasoner, 2 actuators are fully dual redundant, namely the Saab 2000 fbw SHA and the Parker horizontal trim stabilizer EMA. It can also be seen that these two actuators have different response to failure: the SHA have the capability to enter damped mode (Fail-passive) and can fall back to a neutral position (Fail-neutral) while the EMA will enter an frozen state and lock its position (Fail-freeze).

It can also be seen that all actuators except the EMAS EMA and the Parker horizontal stab EMA have a damped mode capability. Interestingly, the J/IST/Moog EHA belongs to some dual-redundant function-classes even though it utilize four components to perform these functions.
In figure 10, the connected components to the two actuators which fulfilled all requirements to be classified as dual-redundant are shown. It can be seen that all functions are conducted with at least two different components, or dual-redundant components. The inherited component functions and attributes are also shown. With this information, the ontology classified these actuators into the defined classes which were presented in figure 9.



Description: EPAD_EHA

Types
- Actuator
- TypeDampedMode
- TypeDeClutching
- TypeEHA_AC
- TypeFreeFloatingActuator
- TypeLinearRetracting

Description: JISTMOOG_EHA

Types
- Actuator
- TypeDampedMode
- TypeDeClutching
- TypeDualRedundantElToRot
- TypeDualRedundantPwrCtr
- TypeDualRedundantRotToHyd
- TypeEHA_DC
- TypeFreeFloatingActuator
- TypeLinearRetracting

Description: EPAD_EMA

Types
- Actuator
- TypeDampedMode
- TypeDualRedundantElToEl
- TypeDualRedundantElToRotVelocitySum
- TypeDualRedundantPwrCtr
- TypeEMA_AC
- TypeJamming
- TypeLinearRetracting
- TypeReversible

Description: EMAS_EMA

Types
- Actuator
- TypeDualRedundantElToEl
- TypeDualRedundantElToRotTorqueSum
- TypeDualRedundantPwrCtr
- TypeEMA_AC
- TypeJamming
- TypeLinearRetracting

Description: Saab2000_FBWActuator

Types
- Actuator
- DualRedundantHydToHyd
- TypeDampedMode
- TypeDeClutching
- TypeDualRedAct_FailNeutral
- TypeDualRedAct_FailPassive
- TypeDualRedundantHydToLin
- TypeFreeFloatingActuator
- TypeLinearRetracting
- TypeSHA

Description: Parker_Horizontal_Stab_EMA

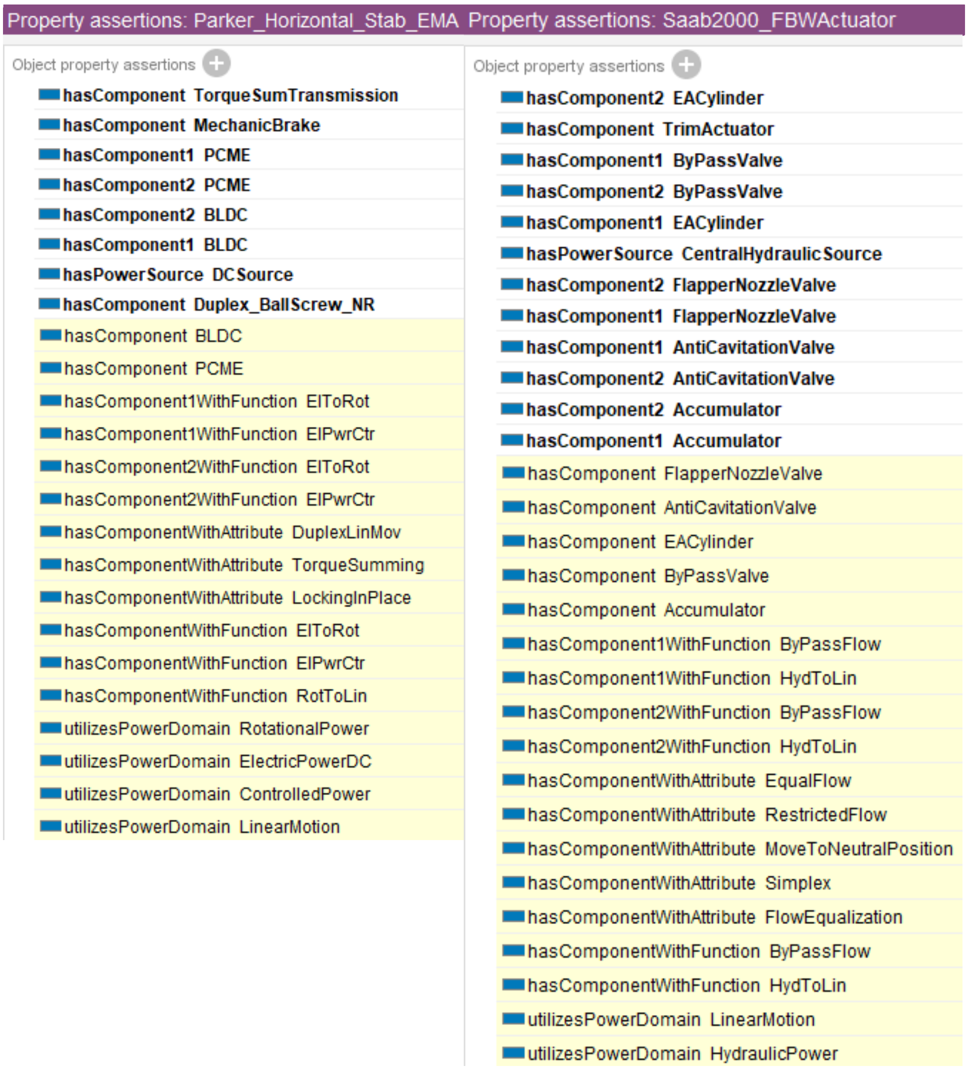
Types
- Actuator
- TypeDualRedAct_FailFreeze
- TypeDualRedundantElToRotTorqueSum
- TypeDualRedundantHydToLin
- TypeDualRedundantRotToLin
- TypeEMA_DC
- TypeNonJamming

Figure 9 – This figure shows which defined classes the different actuators presented in table 4 are inferred by the reasoner to belong in.

4. Discussion

As could be seen by the results, ontologies can sufficiently be used to classify actuators into comparable groups. However, when classifying the actuators used for this case-study, is showed that none of the architectures should be compared against each other with regards to the information and requirements used in this project. The difficulties are to find all attributes and functional requirements which must be set in order to successfully state that the actuators are comparable. It can be discussed whether two actuators which both are dual redundant actually can be comparable, or if all components has to be dual redundant in order to say that the actuator is dual redundant. Actually, redundancy is just a way of ensuring reliability, and a proper classification should thereby ensure that the actuators to be compared fulfill the same reliability requirements. However, for primary flight control actuators, a single fault can never lead to a catastrophic event, and redundancy is one way to mitigate this requirement.

The actuators were not only classified with regards to redundancy of components which perform the different fundamental functions, but also the response to failure, which is an important function of the actuator. It is however difficult to understand the response to failure only by inspection of the actuator components and architecture since the response can, and most often are, performed through software. But, if not the equipped hardware enable a certain response to failure, it can not be performed. To understand what it costs in terms of weight, size and performance to fulfill certain requirements, such as response to failure, with the different actuator technologies, it is important to understand what architectural changes are required to perform the required functions. However, there may be many ways to fulfill such requirements with the different technologies and by including such information in an ontology, an overview of the solution space is created. Together with mathematical models of the involved components, it may become easier to find the most suitable solution. The work conducted here intend to raise the question of how to compare actuators which utilize different technologies,



Property assertions: Parker_Horizontal_Stab_EMA

hasComponent TorqueSumTransmission
hasComponent MechanicBrake
hasComponent1 PCME
hasComponent2 PCME
hasComponent2 BLDC
hasComponent1 BLDC
hasPowerSource DCSource
hasComponent Duplex_BallScrew_NR
hasComponent BLDC
hasComponent PCME
hasComponent1WithFunction ElToRot
hasComponent1WithFunction ElPwrCtr
hasComponent2WithFunction ElToRot
hasComponent2WithFunction ElPwrCtr
hasComponentWithAttribute DuplexLinMov
hasComponentWithAttribute TorqueSumming
hasComponentWithAttribute LockingInPlace
hasComponentWithFunction ElToRot
hasComponentWithFunction ElPwrCtr
hasComponentWithFunction RotToLin
utilizesPowerDomain RotationalPower
utilizesPowerDomain ElectricPowerDC
utilizesPowerDomain ControlledPower
utilizesPowerDomain LinearMotion

Property assertions: Saab2000_FBWActuator

hasComponent2 EACylinder
hasComponent TrimActuator
hasComponent1 ByPassValve
hasComponent2 ByPassValve
hasComponent1 EACylinder
hasPowerSource CentralHydraulicSource
hasComponent2 FlapperNozzleValve
hasComponent1 FlapperNozzleValve
hasComponent1 AntiCavitationValve
hasComponent2 AntiCavitationValve
hasComponent2 Accumulator
hasComponent1 Accumulator
hasComponent FlapperNozzleValve
hasComponent AntiCavitationValve
hasComponent EACylinder
hasComponent ByPassValve
hasComponent Accumulator
hasComponent1WithFunction ByPassFlow
hasComponent1WithFunction HydToLin
hasComponent2WithFunction ByPassFlow
hasComponent2WithFunction HydToLin
hasComponentWithAttribute EqualFlow
hasComponentWithAttribute RestrictedFlow
hasComponentWithAttribute MoveToNeutralPosition
hasComponentWithAttribute Simplex
hasComponentWithAttribute FlowEqualization
hasComponentWithFunction ByPassFlow
hasComponentWithFunction HydToLin
utilizesPowerDomain LinearMotion
utilizesPowerDomain HydraulicPower

Figure 10 – This figure shows the components of the dual redundant actuators. The yellow-marked object property assertions were inferred by the reasoner while all the unmarked ones were manually asserted.

and propose a method of how to classify actuators by only looking at their internal components and functions. By not including sizing and performance aspects, it can be known in an early stage what architectural changes which has to be made to make the actuators comparable before mathematical modeling is conducted.

The actuator which utilized four components to perform different functions was seen to be classified into the dual-redundant class for those functions. This happened since the ontology is now built to include all actuators which surpass the set requirements. This may become a problem when many architectures are included in the ontology and the user is not watchful, actuators designed to fulfill different requirements shall not be compared. However, this can easily be noticed if similar defined classes for quadruple-redundant functions are added. Then, an quadruple-redundant actuator will belong to both this class and the dual-redundant class.

The proposed method is not limited to only use for aerospace actuators, it could be used to classify any type of system which perform a certain function. Generally, when systems are compared, it could be beneficial to map all their functions and attributes in order to fully understand how well the systems meet (or exceed) the functional requirements.

5. Conclusion

As seen in this paper, an approach for classifying actuators without including sizing or performance aspects has been developed. It can be seen that it is possible to classify actuator architectures by using the approach, but difficult to find all required aspects to fully know whether the actuators within a certain group are surely comparable. Further analysis of what is making actuators comparable is required.

When deciding which technology to use for a certain platform, it is important to map the functional requirements of the actuator before selecting. With these, the designer can add more information to the concepts which are to be analyzed and decided on, resulting in a more justifiable comparison and less unknowns when entering an concept development stage. The design changes necessary to comply with a requirement specification may be more or less costly for the different technologies with regards to weight, price and performance.

6. Contact Author Email Address

Mailto: felix.larsson@liu.se

7. Copyright Statement

References

[1] Maré J-C. *Aerospace Actuators 1*. ISTE, St George's Road, London, 2016. ISBN 978-1-84821-941-0.

[2] Qiao G, Liu G, Shi Z, Wang Y, Ma S, and Lim T C. A review of electromechanical actuators for More/All Electric aircraft systems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 232(22):4128–4151, 2018.

[3] Frischemeier S. Electrohydrostatic actuators for aircraft primary flight control-types, modelling and evaluation. *5th Scandinavian International Conference on Fluid Power, SICFP '97, Linköping, Sweden, 28-30 May*, pages 1–16, 1997.

[4] Navarro R. Performance of an electro-hydrostatic actuator on the F-18 systems research aircraft. *NASA Technical Memorandum*, (206224), 1997.

[5] Li J, Yu Z, Huang Y, and Li Z. A review of electromechanical actuation system for more electric aircraft. Beijing, China, October 2016. IEEE/CSAA International Conference on Aircraft Utility Systems (AUS).

[6] Alle N, Hiremath S S, Makaram S, Subramaniam K, and Talukdar A. Review on electro hydrostatic actuator for flight control. *International Journal of Fluid Power*, 17(2):125–145, 2016.

[7] Bildstein M. A built-in jam release device for electromechanical actuators in flight control. *Proceedings of the 6th International Conference on Recent Advances in Aerospace Actuation Systems and Components, Toulouse, France, pp. 105–108, 2–3 April*, 2014.

[8] Baxerres L Todeschi M. Health monitoring for the flight control emas. *IFAC-PapersOnLine 48-21 186–193*, 2015.

[9] Hussain Y M, Burrow S, Henson L, and Keogh P. A review of techniques to mitigate jamming in electromechanical actuators for safety critical applications. *International Journal of Prognostics and Health Management*, 9(12), 2018.

[10] Horridge M, Knublauch H, Rector A, Stevens R, Wroe C, Jupp S, Moulton G, Drummond N, and Brandt S. A practical guide to building owl ontologies using protegé 4 and co-ode tools edition 1.3. *The University Of Manchester*, 2011.

[11] Stevens R, Stevens M, Matentzoglu N, and Jupp S. Manchester Family History Advanced OWL Tutorial. page 76, 2015.

[12] Knöös Franzén L, Staack I, Jouannet C, and Krus P. An Ontological Approach to System of Systems Engineering in Product Development. *Proceedings of the 10th Aerospace Technology Congress, October 8-9, 2019, Stockholm, Sweden*, 162(October):35–44, 2019.

[13] Ast M, Glas M, and Roehm T. Creating an Ontology for Aircraft Design. *Deutscher Luft- und Raumfahrtkongress 2013*, pages 1–11, 2013.

[14] Chourabi O, Pollet Y, and Ahmed M B. Ontology based knowledge modeling for system engineering projects. *Proceedings of the 2nd International Conference on Research Challenges in Information Science, RCIS 2008*, (July):453–458, 2008.

[15] Van Ruijven L C. Ontology for systems engineering. *Procedia Computer Science*, 16(December 2013):383–392, 2013.

[16] Kitamura Y. Roles of ontologies of engineering artifacts for design knowledge modeling. *Design Methods for Practice - 5th International Seminar and Workshop of Engineering Design Integrated Product Development, EDIProD 2006*, DS 37:59–69, 2006.

[17] Bates L B and Young D T. Developmental testing of electric thrust vector control systems for manned launch vehicle applications. *Proceedings of the 41st Aerospace Mechanisms Symposium, Jet Propulsion Laboratory, May 16-18*, 2012.

[18] Jensen S C, Jenney G D, Raymond B, and Dawson D. Flight test experience with an electromechanical actuator on the f-18 systems research aircraft. *19th DASC. 19th Digital Avionics Systems Conference. Proceedings (Cat. No.00CH37126), Oct 7-13, Philadelphia, USA*, 2000.

[19] Thompson K. Notes on "the electric control of large aeroplanes". *IEEE AES Magazine, Dec*, 1988.

[20] Singer G and Persson U. Fly-by-wire for the saab 2000 concept, development and testing. *ICAS-96-3.2.2, ICAS, Sorrento, Sep*, 1996.

[21] Braun R. Hardware-in-the-loop simulation of aircraft actuator. Master's thesis, 2009. ISRN: LIU-IEI-TEK-A-08/00674-SE.

[22] Parker Control Systems. Horizontal stabilizer trim actuator, (accessed: 29.05.2021). https://ph.parker.com/us/en/horizontal-stabilizer-trim-actuator.