UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Study of the movement of an articulated vehicle for agricultural and construction related activities

Document:
Report

Author:
Albert Porta Forradelles

Directors:
Gustavo Adolfo Raush Alviach / Esteban Codina Macià

Degree:
Industrial Engineering Master's Degree

Cursed:
Spring, 2022.

TREBALL DE FI D'ESTUDIS

Abstract

The following report is aimed at describing the creation of a multi solid model of an articulated vehicle. This vehicle is a tractor with dual steering, used in vineyards and similar fields of agriculture which need improved turning and manoeuvring. The main characteristic of the model is that it is parametric, meaning it can be easily modified to simulate different kinds of articulated vehicles. The method used to reach these goals is the theoretical method of Bond Graph, which will be implemented in a simulation software named 20Sim. This software basically does simulations of Bond Graph diagrams, enabling the option of making 3D animations of the modelled systems.

This report aims to describe the basic elements, simple or more complex, of the Bond Graph technique. The scope of this explanation goes from the basic bond graph elements to the more complex building blocks used for the creation of the model. Once it is done, it will be followed by the description of the model of the vehicle.

Even though the simulation of the model with the real hydraulic system was not possible, a simpler one has been used to test the multi solid system, enabling the conclusion that it behaves correctly when actioned with an easily changeable hydraulic circuit. This allows the model to be a test bed for different systems, or iterations of the same one. The main issue with the designed model is that it requires real data to be calibrated, data which was not available during the making of this report.

L'objectiu de la memòria d'aquest projecte és descriure la realització d'un model multi sòlid d'un vehicle articulat. Aquest és un tractor amb direcció dual, el qual s'utilitza en vinyes i àmbits similars de l'agricultura en els quals es requereixen radis de gir petits i una molt bona maniobrabilitat. La principal característica del model és que es tracta d'un sistema multi sòlid paramètric, el qual significa que és fàcilment modificable per a realitzar la simulació de diferents vehicles articulats. El mètode utilitzat per a assolir aquests objectius és l'eina teòrica del Bond Graph, la qual serà implementada en un software de simulació anomenat 20Sim. Aquest software bàsicament realitza simulacions a partir de diagrames fets amb Bond Graph, amb l'opció de realitzar animacions 3D del model realitzat.

Aquesta memòria te l'objectiu de descriure els elements bàsics, simples i més complexes, de la tècnica de modelització Bond Graph. L'abast d'aquesta explicació inclourà des de la descripció dels elements bàsics del Bond Graph fins a la caracterització dels blocs d'elements que s'utilitzaran per a construir el model. Un cop feta, serà seguida per la descripció del model del vehicle.

Encara que la simulació del model utilitzant un sistema hidràulic real no ha estat possible, un de més simple s'ha fet servir per a provar i validar el sistema multi sòlid, permetent arribar a la conclusió que aquest es comporta adequadament al ser accionat per un circuit hidràulic. Això habilita al model el poder ser utilitzat com a banc de proves per a diferents sistemes, o iteracions d'un mateix. El major problema amb el model dissenyat és que requereix de dades reals per a ser calibrat, dades que no han estat disponibles durant la realització del projecte.

Acknowledgements

I would like to express my gratitude to Dr Germán Filippini for his valuable and constructive lessons on the Bond Graph method in three dimensions. Lessons without which the realization of the goals of this project would not have been possible.

I would also like to express my gratitude to both my tutors, Dr Esteban Codina and Dr Gustavo Raush, for their help and guidance during the project, and their valuable advices and critiques of this work.

Finally, I would like to thank my family for their support and encouragement throughout my study.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Index

Table Index

Figure Index

1. Introduction

1.1 Object

The object of this study is to analyse the behaviour of an articulated vehicle, which is used normally on agriculture and construction, especially on operations where space is reduced and small turn radius are needed. To study the behaviour a multi solid system a simulation will be done, incorporating also the corresponding hydraulic system for this types of vehicles.

1.2 Scope

The scope of this project consists in the creation of a model to do the simulation of the behaviour of a vehicle, with a central articulation and front wheel steering. To achieve this objective, firstly a formation on the theory of the dynamics of rigid solids will be done. Continuing with the necessary formation, the theoretical tool of Bond Graph will be explored, both in one and three dimensions. The first one is used to model the hydraulic system, and the second, to model the dynamics of the vehicle. Finally, in terms of the learning phase to face the challenges of the project, a formation with the software 20Sim, which is based in Bond Graph modelling, is required.

The next point to be addressed of the project is to make a description of the vehicle, or machine, to model. All of the technical characteristics will need to be mentioned; its performance and dimensions especially. This description will be of use later in the project to create the simulation model.

To create a model which is as trustworthy as possible, an analysis of articulated vehicles will be performed. The objective of this analysis is to identify the different important sets of rigid solids which conform the machine, aiding in the creation of a simplified model which adjusts to reality as accurately as possible.

Once the machine is described and the system of solids that conforms it is identified, the next step will be to proceed to create a model with the Bond Graph in 3D technique. The model to create will also include the simulation of a hydraulic system done with Bond Graph 1D. This system will control the steering of the vehicle. The model will be validated with a comparison with experimental data if possible.

1.3 Requirements

To obtain an adequate model this series of requirements will need to be followed:

- The model has to be made in a general manner, meaning that it has to be valid for a variety of articulated vehicles, even though it will be based in a concrete one. The different parameters that define the vehicle should be able to vary with the objective of making the simulation of a different articulated vehicle.
- The model has to be created with the Bond Graph multi solid modelling tool.
- The software used to obtain the model has to be 20Sim.
- Modifications to the hydraulic system have to be performed easily, without significantly affecting the dynamic model, to experiment with different circuits. This requirement is aimed to allow experimentation with the hydraulic control of the steering.
- The vehicle model has to be accompanied by an animation that complements the numeric results obtained in the simulation.

1.4 Justification

Due to the increase in the computational capacity of computers, the doors have been opened to the use of a variety of techniques to simulate reality. This revolution has led to the development of software capable of simulating in a precise manner the different behaviours of the real world, which professionals of different fields encounter on their day to day work. Whether in trying to predict the behaviour of society, planning public transport itineraries or making structural calculations, this revolution in the informatics world has led to the creation of tools that significantly facilitate the work of professionals, who have to face the physical and sociological challenges that are encountered in their everyday work.

This revolution has affected significantly in the engineering field. It has done it through the creation of software capable of solving complex problems with a relative ease, principally due to the great computational capacity of the new computers, especially when considering super computers. Even though this increased capacity of processing data and problem solving has improved and eased the work of professionals, it has to be always accompanied by a greater capacity of result interpretation, to ensure the solutions obtained are correct.

Due to the previously exposed advances, the last few decades a new branch in engineering has been established, simulation. Thanks to the increased computational capacity, now it is possible that the majority of engineers are capable of checking the results of their calculations through simulation. Obviously there are fields of engineering where the computational capacity required is higher than the one that can produce a personal computer or a laptop, but in the majority of cases the results obtained with simulations in one of both is enough to validate the calculations or design.

The main advantage of simulation is its cost, which is significantly smaller than the one required to do experimentation. The requirement for real experimentation is still primordial, but when it is combined with simulation, the creation of new designs and calculations is streamlined and made cheaper. In the creation of any model to analyse something or to experiment, firstly a calibration is made using real data obtained through experimentation.

In a field of engineering where simulation has become really useful is in the studying of solid systems in movement. Whether the studied system is a mechanism of a machine or a vehicle, making their design and validating it is a complex process which requires iteration. It is costly when it entails the creation of prototypes, especially when they are at real scale. To streamline this process and make it cheaper a digital model is incorporated to validate the design made by the team of engineers, making a fast and cheap iteration process. This incorporation minimizes the number of prototypes and the modifications made to them.

In the case of this project, the realization of a model of an articulated vehicle can help engineers in the design of the hydraulic system for controlling the steering. Through the created model, obtaining an approximation of the behaviour of the vehicle, for different variations in the parameters and layout of the hydraulic system, will be possible. The engineers will be able to incorporate different loads and modify the terrain parameters with the objective of validating the hydraulic design for the steering of the vehicle.

Since the model will be parametric, meaning all the values of the model will be easy to modify, the experimentation with a variety of hydraulic circuits, with different components and layouts, will hopefully aid in creating an optimal solution for the steering system of an articulated vehicle.

2 Antecedents and State of the art revision

2.1 Steering typologies

To control de course or the orientation of a vehicle or a mobile machine a steering system is required. It has to be designed to adequate correctly to the field of work for which the machine/vehicle has been designed. Depending on the purpose, a steering system and its control will have to be designed for it to comply with the requirements imposed by the work environment. Due to the great variety of work situations, different types of steering exist:

2.1.1 Front steering

This type of steering is the most common, and is the one to which most of citizens are used to. Whether in the majority of commercial vehicles, like cars or motorcycles, or most of transport vehicles, like trucks or busses, we find that they've been designed with front wheel steering.

Figure 1. Tractor with front wheel steering

This type of steering has also been used in most of agricultural tractors and industrial transport trucks. It is often used in those vehicles or machines which have to reach a moderate to high speed, even though in some cases is not the only type. The main advantage of this steering type is the great stability and equilibrium it provides while steering at high speeds. It describes an angle of rotation that is comfortable and also intuitive for the driver.

In heavy vehicles, whether they are in the fields of construction or defence, we can find lots of cases that have been designed with more than four wheels. Due to the great dimensions of this vehicles, sometimes steering is added to more than one pair of wheels to aid with manoeuvring.

In the case previously described is common to add rear wheel steering to help with the manoeuvring of the heavy, and usually large, vehicles. This combination is also often used in high performing cars. A subtle rear wheel steering is added to improve turning in high demanding situations, like in a race track.

2.1.2 Rear wheel steering

This steering type is really common with mobile machines, whether they operate in the industrial, agricultural or construction fields. Rear wheel steering is typical in forklifts, vehicles used in a great variety of fields, but especially on the industry in the role of transport and management of goods and stock within warehouses. Another important example of rear wheel steering are cereal harvesters. This machines are the go to for recollecting a wide variety of crops, being most famous for cereals and corn. Since they operate with a cutter which is wider than the machine, having a good steering radius facilitates its operation on the field.

Rear wheel steering is really useful due to the great capacity of turning that it provides to the machine that mounts it. It can achieve almost 90 degree steering angles, which makes it ideal for machines that have to operate in reduced spaces, like the previously mentioned forklifts. The high turning angles are also ideal for machines that need to manoeuvre describing tight radiuses, like the harvesters or dumpers for construction.

Another important reason of the use of rear wheel steering is related with weight distribution in a machine. In most machines a significant weight is positioned in the front wheels, so a bigger and more capable steering system would have to be installed. Having rear wheel steering also decreases inertia moments, since with this type of steering the charge supported in the front travels slower than it would with front steering.

Figure 2. Harvester with rear wheel steering

The most important disadvantage of this steering type is that it can end up being dangerous at high speeds, mostly due to that it can provoke turns that make the vehicle loose its stability. As said previously in the front wheel drive part, heavy vehicles can mount steering in more than one pair of wheels.

2.1.3 Central/Articulated steering

When the performance required of a vehicle or machine make it have dimensions that complicate some of the operations for which it has been designed, normally engineers resort to the implementation of a central articulation that make the machine able to modify its geometry. This modification will allow the vehicle to reach higher turning angles, enabling the machine to perform manoeuvres that would be impossible due to its big dimensions.

Figure 3. Dumper with central steering

This type of steering can be found in dumpers, whether they operate in construction or in mining. On agriculture the previously described problematic can also be found, this is why some tractors, normally of small dimensions, also include a central articulation with its corresponding steering system.

This type of steering is often used as the main steering system in vehicles that operate in enclosed work environments as mines or construction operations. In other environments usually a frontal steering system is added. This allows for extremely tight turning radiuses, giving the operator a lot of freedom for manoeuvring.

2.1.4 Tracked steering

For vehicles , or machines, which have been designed to manoeuvre or circulate in difficult terrains, where is needed that they have a great capacity of traction, designers usually opt to use tracks, which depending on the use can be made of rubber or metal. This tracks provide a great traction capability and stability to the vehicles, which in most cases have a high mass. The steering is accomplished by varying the relative speeds of each of the two tracks to orientate the vehicle.

Figure 4. Tracked excavator

This type of steering is used normally in vehicles that operate in construction, mining and also in some fields in defence. The most typical examples of tracked vehicles are excavators and tanks. It can also be found in agriculture, especially in the heaviest and better performing equipment, but normally the tracks are used for their higher traction capacity, and the steering is done with one of the previous types.

2.1.5 Combinations

The different types of steering described previously can be, in most cases, combined to achieve the manoeuver capability required for the designed machine. The most common combinations have already been described, but in niece applications in the industry and defence more extravagant and complicated steering combinations can be found.

2.2 Multi solid system analysis techniques

2.2.1 Newton

For the analysis of multi solid systems is really common the use of Newton's second law [1]:

$$F = ma\,; \tag{1}$$

To completely study a system the angular moment components need to be added, so the equations system to be solved will consist of six equations.

$$F(x, y, z) = ma(x, y, z)\,; \tag{2}$$

$$M(\theta, \tau, \varphi) = m\alpha(\theta, \tau, \varphi)\,; \tag{3}$$

Using this equation system the behaviour of a multi solid system can be obtained, of course this is possible with a maximum of six unknown variables.

From Newton's second law various, more complex, methods like Bond Graph have been created. This method consists in forming diagrams which are made based on that flux of energy that exists between the different elements of a system. Bond Graph is a really useful tool to model various mechanical systems [2] [3].

2.2.2 Euler-Lagrange

Another way of formulating Newton's second law, and consequently another method for solving multi solid systems is de use of Euler-Lagrange equations [1]:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt}\frac{\partial L}{\dot{q}_i} = 0\,; \tag{4}$$

In this equation the Lagrangian term appears:

$$L = T - V\,; \tag{5}$$

Where T is the kinetic energy and V the potential one. We also find the term q in the equation, which represents a vector that defines a point in space.

2.2.3 Hamilton

The Hamilton equation is a reformulation of the Euler-Langrange one, which consists on the substitution of the velocity vector,$\dot{q}_i$, for the one that defines the solids momentum p [1]. The obtained equation is as follows:

$$\frac{dq}{dt} = \frac{\partial H}{\partial p}\,; \tag{6}$$

Where H is called a Hamiltionian:

$$H(p, q, t) = \sum_{i=1}^{n} p_i \dot{q}_i - L\,(q, \dot{q}, t)\,; \tag{7}$$

And the momentum is:

$$p_i = \frac{\partial L}{\partial \dot{q}_i}\,; \tag{8}$$

2.2.4 Comparison of the three methods

	Newton	Lagrange	Hamilton
Equations	$F = ma$	$\frac{\partial L}{\partial q_i} - \frac{d}{dt}\frac{\partial L}{\dot{q}_i} = 0$	$\frac{dq}{dt} = \frac{\partial H}{\partial p}$
Coordinate system	Cartesian	Independent of any particular system	Canonical
Advantages	Easier to formulate than Lagrange and Hamilton	Improved flexibility in respect to Newton due to not being dependent on a single coordinate system. The coordinates that make the equation easier to solve can be chosen.	Allows the conserved quantities to become more apparent.
Disadvantages	Hard to solve in complicated systems or ones that have a large numbers of interacting objects.	More complex formulation than Newton.	More complex to formulate than Newton and Lagrange
Area of application	Simple and intuitive systems	Complex systems	Complex systems if symmetries and conserved quantities are relevant to the model

Table 1. Classical mechanics methods

To summarize, Newton's second law is mainly used to solve multi solid systems that are on the simpler side. This is because using this method creates a big number of equations, which raises simulation time significantly. The more solid bodies, and interactions between them, the more complex and hard to solve the equation system becomes. Because of this, a new formulation was created from Newton's second law. The Lagrange method, although more complex to formulate, enables the modelling of multi solid systems with a smaller set of equations than Newton. This is because these equations can be accommodated to different coordinate systems that better solve the sub-assemblies within the model. This is why this method is used in most multi solid simulation software. Finally there is Hamilton mechanics, which formulates the different equations centred around the conserved quantities of the mechanical system. This formulation is more complex than the Lagrangian mechanics, so this method is not normally used as much.

2.3 Software for multi solid modelling

2.3.1 ADAMS

The ADAMS software consists in a set of programs dedicated to the modelling and simulation of a great variety of study cases in the engineering world. The principal fields where this software is used are: aerospace, automotive, defence, electronics, energy, machinery, medicine, railway systems and naval sectors [4].

Adams™

Figure 5. ADAMS TM logo

In the set of software that the company offers a series of dedicated programs to the different stages of design, of the fields mentioned previously, are available. The main software is the one dedicated to the simulation of multi solid systems, which is complemented by software that integrates flexible solids, components that incorporate the ability to obtain resistances to fatigue and stress, analysis of vibrations in the system and real time simulation.

This software is one of the leaders in the field of simulation of multi solid systems, where a big competition between companies exist. The supplying company is MSC Software, and the set of programs have been recognised by entities like Boing, NASA, and others. This software is bought under license, the price of which depends on the set of complements and the number of PCs that will operate it.

The software use consists in the generation of a 3D model, binding the different components with the corresponding union systems. This can be done by importing the solids or creating them with the ADAMS software. The resolution of the multi solid system is usually accompanied by a 3D animation. To solve the systems the Euler-Lagrange equations, which have previously been described, are used. The system to be solved is formed by algebraic and second order differential equations, which require sophisticated numerical systems to be solved [5].

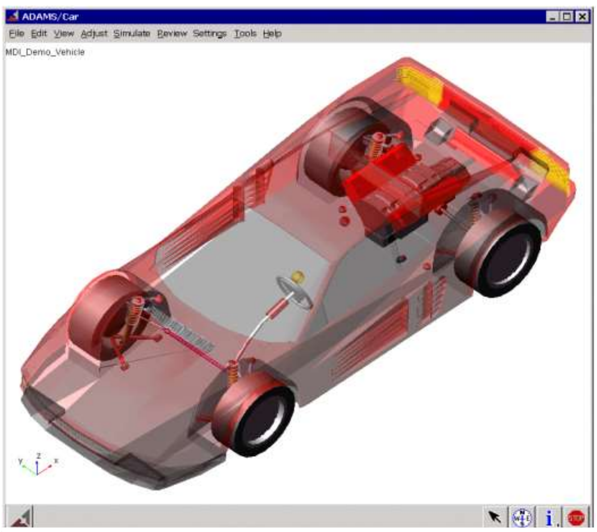


Figure 6. ADAMS Car model

2.3.2 20Sim

20Sim is a software created for the simulation and modelling of various systems, including multi solid ones [6]. This program differentiates itself from some of the leaders of the field in that it works through the graphical representation of the models, generally with the theoretical tool of Bond Graph, but the possibility of coding different blocks also exists. The company compares the necessary graphic to create the model with an engineering drawing made to represent a problem or situation.

Figure 7. 20Sim™ logo

The fields of use where 20Sim is used are similar to the ones which have been previously mention in the ADAMS section: aerospace, automotive, machine and naval sectors. As we can see the application fields are fewer, mostly due to a reduced number of components in the software.

The differentiation factor of this software is that it was the first one to use the Bond Graph system. It first started in 1995, so if working with Bond Graph is preferred, this is one of the more ideal programs to use.

Bond Graph allows the making of a physical systems model, which tries to be as close as possible to the real one. This model is made from a set of imaginary elements which are arranged to create a representation of the physical reality, whether this reality is a set of solids, a pneumatic or hydraulic system or an electric one. This flexibility exists because Bond Graph is based in the conservation of energy, so the method can be applied in a variety of fields, in some cases outside of physics and engineering.

As previously mentioned in the section of theoretical tools for the analysis of multi solid systems, Bond Graph is based on Newton's second law to solve multi solid systems. The result is the generation of more equations than the other methods, but this ones are easier to solve. This software also includes a 3D animation tool that allows the visualization of the behaviour of the system.

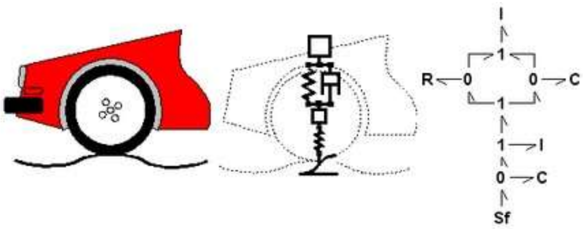


Figure 8. Bond Graph example

2.3.3 Open Dynamics Engine

Due to the mention in this section of two commercial software for the simulation of multi solid systems, is also necessary to incorporate to this list an open software.

Open Dynamics software is created to simulate articulated rigid solids, which are bounded together with advanced joints, and can have friction and collision detection. Even though is used in video game creations, this software can also be of use in the modelling of multi solid systems as the one that will be analysed in this project. The software requires a moderate level of C/C++, and like ADAMS it uses the Euler-Lagrange equations [7].

3 Direction type and software election

As previously mentioned, the objective of this project is the creation of a multi solid model to make simulations of a tractor that has to operate in vineyards. Due to this restriction, the vehicle requires a great turning capability. This requirement is necessary mainly because of the distribution of the crops of grapes, which are organized in straight lines with small spaces between them. This distribution requires the tractor to describe really closed turns. This leaves two realistic options to be able to make the necessary manoeuvres to operate in those fields: a rear wheel steering system or the combination of front and central steering.

In order to make this decision the jobs that the vehicle will perform have to be taken into account. These are basically to chisel the land between the vines, spray fertilizers, and carry the collected grapes during the harvesting season. Those actions have a thing in common, the machinery used with the tractor is mounted in the rear side of the vehicle. For this reason a rear steering system would make difficult the manoeuvring operations, and this is why most tractors for vineyards and similar crops use a front wheel steering system. The objective is to maximise the manoeuvring capability, so adding a central articulation would decrease the turning radius without significantly affecting the trajectory of the mounted machine, avoiding collision with the crops. For this reasons this type of steering system would give the tractor a great manoeuvre capability without significantly affecting its ability to mount most commercial machines.

In regards to the software to be used, the main reason for its choice is the capacity of obtaining it to make this project. Firstly, the Open Dynamics Engine option is discarded, mainly due to the required formation needed to use the software and obtain the standard of results required. A longer formation period would be necessary with ODE in comparison with the first two options. The election between both commercial software, ADAMS and 20Sim, is conditioned basically by the acquiring of the academic license, which has been easier to obtain with 20Sim.

Even though ADAMS is more intuitive, mostly due to the making of the model directly with 3D objects, the use of 20Sim and the Bond Graph system gives a better understanding of how the model works, whether it is a hydraulic system or a car. This is due to the way in which the model has been created, using the same basic elements to form the different components, and aiding in the understanding of the relations between them, giving better knowledge about the created model. It could be summarised by saying that with 20Sim the model is created in an inferior layer than the use of the 3D blocks/components.

The main disadvantage of using 20Sim is the requirement of a previous formation in the Bond Graph system, without it, it would not be possible to create the multi solid model of the tractor. Due to the necessity of making a Bond Graph graphic, more time will be dedicated in some of the steps needed to make the model, mainly the one that consists in making a simplification of the vehicle. This step aims to minimize the number of solids to facilitate the creation of the model.

The software is in general less intuitive, from de generation of the model to the making of the 3D animation, but the possibility of obtaining the license, the previous knowledge of the tutors in the Bond Graph theory, especially in the hydraulics field, has led to the choosing of 20Sim as the program to develop this project. The software, once done the formation in both Bond Graph and 20Sim, will allow the creation of the desired model and its parameterization, to make it valid for similar vehicles or machines.

4 Vehicle description

To continue with the design decision made previously, which consisted in choosing a tractor with front steering and a central articulation, a concrete model of a vehicle is needed in order to continue with the study. Since the type of tractor is not very common, there are a reduced number of manufacturers that supply this type of machine. Even though the number of brands might be higher, there are few manufacturers. One of the main European manufactures of articulated tractors for the specific work environment is called BCS [8].

BCS is an Italian manufacturing company that specialises in small articulated tractors and other machines. We can find them in brands such as Ferrari, where different versions of articulated vehicles can be found. For this project the exact version of tractor is not really important, since the objective is to make a model that can be adapted to different articulated vehicles. For this reason, a model is chosen based on the available information regarding geometry and weights.

The model chosen is the Ferrari Vega 85. This is one of the more capable machines of the brand, being of course one of the biggest and more powerful. The tractor is offered with or without a cabin, and when it does not have one it includes a roll bar for safety in case of a rollout. The main characteristics that differentiate it from other tractor is the central steering and the isodiametric wheels.

Figure 9. BCS Ferrari VEGA 85

The main specifications of the tractor are as follows:

- It is powered by a 75.3 horsepower Khole engine, specifically the KDI 2504 TCR Turbo with four in line cylinders and a displacement of 2482cc. It makes peak power at 2300 rpm and peak torque between 1000 and 1800 rpm. It is considered one of the top common rail engines in the market.

Figure 10. Khole KDI 2504 TCR Turbo

- The tractor has a Power Clutch system that allows the operator to shift without pressing the clutch with his foot. In case of driving with the clutch pedal, it has an aid system to facilitate changing gears. It also includes the engaging and disengaging of the clutch with the brake pedal for braking and stopping situations.
- The power of the engine is transferred with a 4 range and 4 velocities gearbox. This gives the tractor 16 velocities forwards and 16 backwards, with speeds ranging from 70 m/h to 40 km/h
- The overall dimensions in millimetres of the vehicle, with a width of approximately 1350 mm, are as follows:

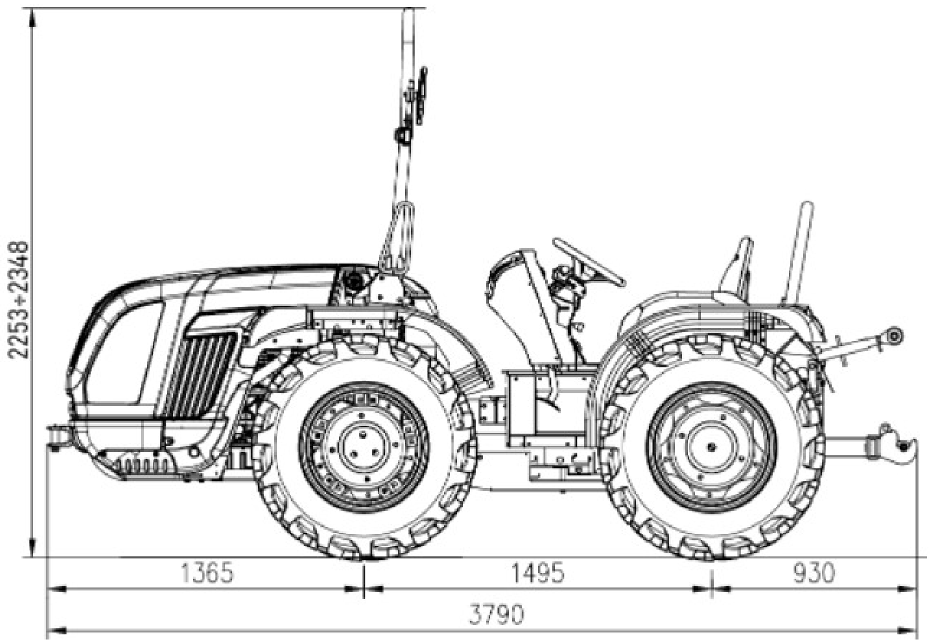


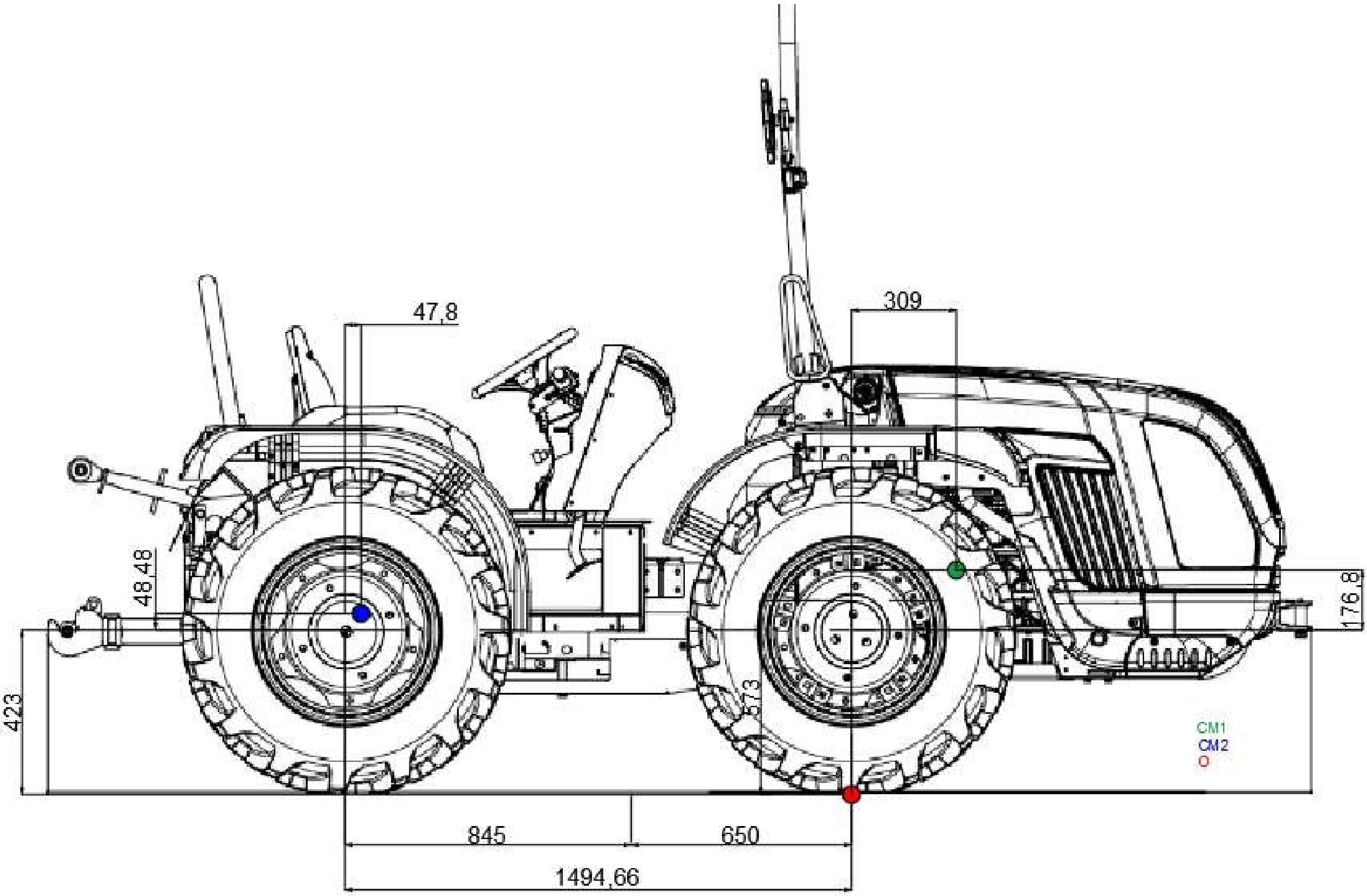
Figure 11. Ferrari VEGA 85 dimensions

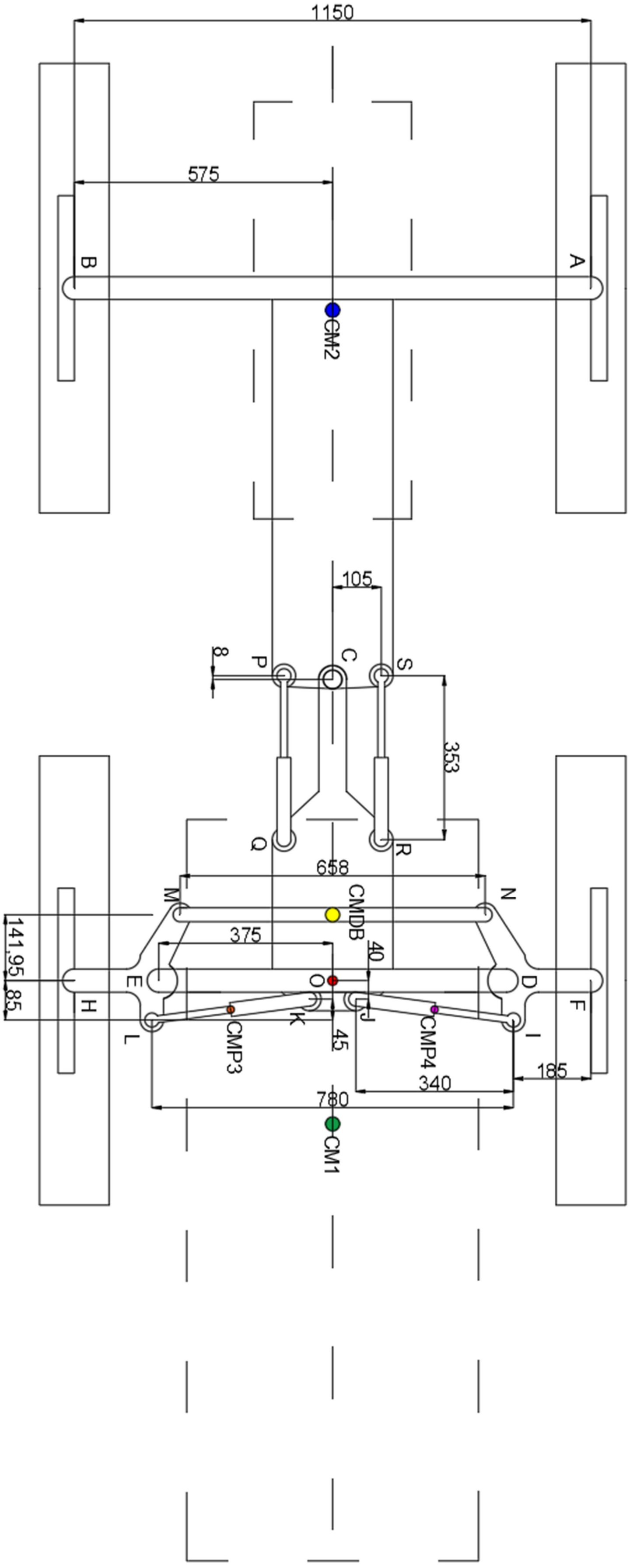
Even though an overall description has been given, it is not enough for the creation of the model. An identification of the main solids of the model, and the main points or links between them is vital to enable the creation of a representation that is as accurate as possible. The aim of this part of the project is to give a list of all the geometrical points of interest with their x, y and z coordinates. To achieve this, the simplified drawings of the tractor have been made.

The first step before the interest point identification is to simplify the tractor in a select set of solids, in order to aid in the selection of points. Since the tractor is an articulated vehicle it will have two main solids and another four representing the wheels. This is an acceptable description that both simplifies the vehicle and gives an accurate geometrical and dynamic representation. This description is valid because the tractor does not have suspension, the axels are solid mounted to their respective side, enabling the simplification of the main frames into two solids.

Even though a simplified description of the vehicle has been given, there are solids missing in order to enable the model to steer. Firstly two solids exist, one on each side of the frontal axel, to enable the wheels to turn. To this solids the piston and the steering bar are mounted, completing the steering system. This solids are also hubs where the front wheels are mounted.

As mentioned before, pistons are also used, and 2 solids are required for each, one for the tube/barrel and the other for the piston rod. Given there are 4 pistons, this gives an additional 8 rigid solids. Finally another solid is required for the steering bar, which is used to keep both wheels pointing in more or less the same direction. The approximate drawing that describes the location of the different interest points for the model is the following:




1150
575
B
A
CM2
105
8
P
C
S
353
Q
R
658
M
N
CMDB
375
40
141,95
85
E
O
D
H
F
K
J
L
I
CMP3
CMP4
45
340
185
780
CM1

To create the list of points, a coordinate's origin is selected. This point is located in the line between the centres of the tractor wheels, which are the front ones, the floor and the centre of symmetry in the longitudinal direction. This point is marked in the drawings in red. The different interest points are found in the following list:

	Position (mm)		
Rigid Solids CM	X	Y	Z
V1 (CM1)	309	0	599.6
V2 (CM2)	-1447.2	0	471.5
VBR (A)	-1495	575	423
VBL (B)	-1495	-575	423
VFR (F)	0	575	423
VFL (H)	0	-575	423
VAR	0	390	573
VAL	0	-390	573
P1A (Q)	-305	-105	573
P1B (P)	-658	-105	573
P2A (R)	-305	105	573
P2B (S)	-658	105	573
P3A (CMP3)	62.5	-220	573
P3B (CMP3)	62.5	-220	573
P4A (CMP4)	62.5	220	573
P4B (CMP4)	62.5	220	573
DB (CMDB)	-141.945	0	573
Union Points			
P12 (C)	-650	0	573
P1FR (E)	0	-375	423
P1FL (D)	0	375	423
P2BR (B)	-1495	-575	423
P2BL (A)	-1495	575	423
PTFR	0	-575	0
PTFL	0	575	0
PTBR	-1495	-575	0
PTBL	-1495	575	0
P1AR (E)	0	-375	423
P1AL (D)	0	375	423
PS (S)	-658	105	573
PP (P)	-658	-105	573
PR (R)	-305	105	573
PQ (Q)	-305	-105	573
PJ (J)	40	50	573
PK (K)	40	-50	573
PI (I)	85	390	573
PL (L)	85	-390	573
BD1 (N)	-141.945	329	573
BD3 (M)	-141.945	-329	573

Table 2. Point positions

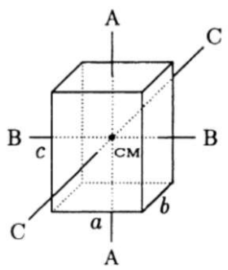
The different point's position can be found in the drawings to easily identify where they are, but the list is important for future steps in the modelling process. The same can be said for the name of the points, even though is better to use just one letter, another nomenclature is used in order to better interpret some of the points:

- The wheels are classified by their position, back left (`BL`), back right (`BR`), front left (`FL`) and front right (`FR`).
- The union points are set with a letter `P` followed by the name of the two solids that are joined.
- In the case of the pistons the drawings letter is used and for the steering bar the `BD` letters are used.
- In the case of `PT`, is the contact point of the tires to the ground.
- In the case of the pistons `A` is the barrel and `B` is the rod. In the front steering the centre of mas is put in the half point of the pistons, and in the central steering the centre of mass is put in the joints with the main solids `V1` and `V2`. This is done to facilitate giving the initial angle to the front pistons.

Before moving to the creation of the Bond Graph, it is important to obtain an approximate value for the inertia of each of the 17 solids. In the case of the lighter ones their value is not considered since it won't significantly affect the dynamical behaviour of the model. The wheels inertia moment is calculated as if they were solid cylinders, and for the two volumes that conform the tractor their inertia is approximately calculated as if they were a parallelepiped or a "box".

The values of inertia moments used in this project are approximations given the data available, but in a design project usually these data is obtained with the CAD software used to design the different parts of the vehicle. With the used method of obtaining the inertia of the different solids the model will be accurate enough for an academics project, but having the inertia calculated with a CAD gives better and more accurate values. The equations used are the following:

- For `V1` and `V2`:



Rectangular Prism

$$I_{AA} = \frac{1}{12} m\left(a^2 + b^2\right)$$
$$I_{BB} = \frac{1}{12} m\left(b^2 + c^2\right)$$
$$I_{CC} = \frac{1}{12} m\left(c^2 + a^2\right)$$
$$V = abc \qquad (9)$$

- For the wheels:

$$I_{central\ axis(y)} = \frac{1}{3} m r^2 \ ; \qquad (10)$$

$$I_{central\ diameter(x,z)} = \frac{1}{2} m r^2 + \frac{1}{2} m b^2 \ ; \qquad (11)$$

Where r is the radius and b is the width of the wheel.

In Table 3 we can find the values of inertia given to the various solids and their mass.

Rigid Solid	m	Ix	Iy	Iz
V1	1210	127.23	338.93	291.75
V2	650	41.62	87.12	62.83
VBR	80	5.02	8.84	5.02
VBL	80	5.02	8.84	5.02
VFR	80	5.02	8.84	5.02
VFL	80	5.02	8.84	5.02
VAR	20	0.07	0	0.07
VAL	20	0.07	0	0.07
P1A	10	1	1	1
P1B	10	1	1	1
P2A	10	1	1	1
P2B	10	1	1	1
P3A	10	1	1	1
P3B	10	1	1	1
P4A	10	1	1	1
P4B	10	1	1	1
BD	10	1	1	1

Table 3. Mass and moments of inertia

We can see that for the steering hub solids (VAR and VAL) and for the pistons and steering bar the inertia moment is considered not relevant since their weight is really small in comparison to the rest of the vehicle. The forces that the pistons will be subjected to will be mainly due to the supports to the other solids. Another reason to declare the inertia not relevant is that the speed of the pistons will not be quick enough for it to make inertia moments relevant.

In summary, there are 6 types of rigid solids that can represent the whole tractor. The first types are the two main volumes that will be joined by the central articulation, V1 and V2. Another important type of the models rigid solids is the articulated hub, where the hub of the wheel is mounted, whose function is to rotate the front wheels actioned by the pistons. The name of this solids are VAR and VAL, for Volume Articulated Right/Left. The other important type are the wheels, which there are four, VBR, VBL, VFR and VFL, for Volume Front/Back Right/Left. Finally there are the pistons P and the steering bar BD. In the pistons A is the barrel and B is the rod.

Once all the solids are defined and their characteristics obtained the next step is to move on to the creation the model, which is done by joining them and applying the necessary restrictions to obtain the desired behaviour of the studied tractor.

5 Basic elements of the model

Bond Graph is a graphical method for representing physical systems with a set of imaginary elements, each representing a physical characteristic. In this part of the project this set of elements and the overall operation of Bond Graph will be explained, with the main goal of describing the main building blocks of the model, which will be composed of these imaginary elements. Since Bond Graph consists in representing the flow of energy between different elements, it is important to explain how this flow goes from one element to another. Each union consists of an effort and a flow, and depending on the causality of the union one of them will go on one direction and the other in the opposite one. Firstly, a list of efforts and flows from different physical systems is made in Table 4 [9].

Physical field	Effort	Flow
Mechanical	Force (N)	Velocity (m/s)
Mechanical	Torque (Nm)	Angular speed (rad/s)
Fluid mechanics	Pressure (Pa)	Flow (m^3/s)
Electrical	Voltage (V)	Current (A)

Table 4. Types of effort and flow

5.1 Power transfer element

This efforts and flows go from one element to another, one in the opposite direction as the other, and this is represented by the symbol in Figure 12 [10].

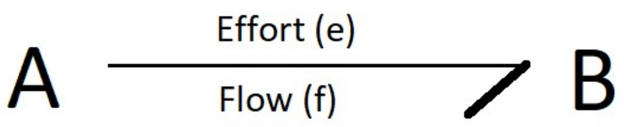


Figure 12. Graphical union element

As seen in Figure 12, the union between two elements will send two signals, e and f, in opposite directions. If the effort goes from A to B, then the flow will go from B to A. How this happens is determined by the causality of the union, which is represented by a vertical line at the end of the arrow. Work exists to optimize the causality in a diagram to optimize simulations [11].

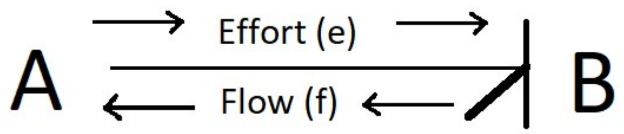


Figure 13. Causality example 1

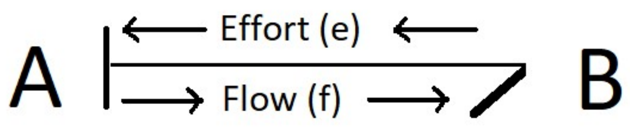


Figure 14. Causality example 2

During the realisation of the model another linking symbol will be used, which is a simple arrow that will carry a single signal, not two like in the mentioned case.

5.1.1 3D power transfer element

Most of the power transfer elements that will be used in the creation of the model will not be 1D like the ones previously described, which have only 2 signals. The transfer element used will consist in two 3 dimension vectors, one for the effort and the other for the flow. This 3D element will be represented with an extra line.

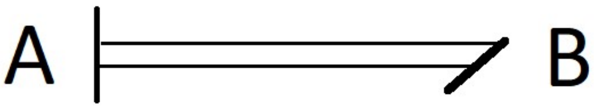


Figure 15. 3D power transfer element

5.2 Source elements

As mentioned previously, the Bond Graph diagrams are based in the transmission of a flow and an effort, so there exist two elements that introduce a value of this two types of magnitudes. These elements are called source elements [10].

$$Se$$

$$Sf$$

With this two elements, since they introduce an effort and a flow, they both have a fixed causality. In the case of the flow the bar is on the side of the source, and in the case of the effort the bar will be set on the opposite site.

There exist a variation of this sources which are modulated. In some cases it will be necessary to control or vary the values, so a signal can be sent to the modulated source to control it.

$$MSe$$

$$MSf$$

5.3 Passive element

There exist another types of imaginary elements to aid in the representation of a Bond Graph model. This type is called passive elements, and it usually represents friction in mechanical systems and resistances in an electrical one [10] [13].

$$R$$

The equation that governs this element is as follows:

$$p.e = f(p.f)\,; \tag{12}$$

$$p.e = r * p.f\,; \tag{13}$$

Where `p.e` is the effort value and `p.f` the flow value that enters the R element. This returns a proportional effort to the flow that has entered the element, or a loss of flow proportional to the effort, depending on the causality.

5.4 Active elements

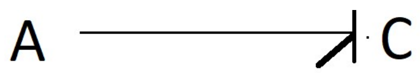
There exist two types of active elements in a Bond Graph diagram [10].

5.4.1 Capacitor

This type of element is used to represent a spring in mechanical systems, a capacitor in electrical systems and compressibility in other systems [13].

$$C$$

Depending on the causality, the equations used in this element are as follows:



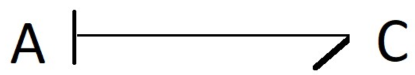
In this case:

$$p.e = cx\;; \tag{14}$$

$$x = \frac{p.e}{c}\;; \tag{15}$$

$$\frac{dx}{dt} = p.f\;; \tag{16}$$

And in the opposite causality:



The equations are as follows:

$$x = \int p.f\;; \tag{17}$$

$$p.e = cx\;; \tag{18}$$

In both cases it can be seen that this equations are the same as the ones of a spring, so in the dynamic model this capacitor will represent the compressibility of an element.

5.4.2 Inertia

To represent a physical Bond Graph model a basic and important element to include is inertia [13]. This element has a mass assigned to it:

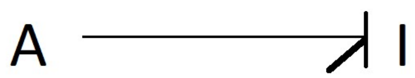
$$I$$

The equation that this element uses is as follows:

$$\frac{p.e}{m} = \frac{dp.f}{dt}\;; \tag{19}$$

$$p.f = \int \frac{p.e}{m} dt\;; \tag{20}$$

In its usual causality:



5.5 Transformation elements

This type of elements are used to alter the magnitudes that flow between the different Bond Graph components. There are two types of transformation elements [10].

5.5.1 Transformer

The transformer element is used to change the magnitude of the energy flow. For example, if force and velocity are used, by multiplying them with a surface value, pressure and flow is obtained.

$$TF$$

$$p.e_1 * TF = p.e_2 ; \qquad (21)$$

$$p.f_1 * TF = p.f_2 ; \qquad (22)$$

5.5.2 Gyrator

A gyrator is also used to change the magnitude like the transformer, but in this case the resulting flow depends on the entering effort, and the resulting effort depends on the entering flow:

$$GY$$

$$p.f_1 * GY = p.e_2 ; \qquad (23)$$

$$p.e_1 * GY = p.f_2 ; \qquad (24)$$

This can be used to represent in Bond Graph effects like Coriolis.

5.6 Union elements

To join all the previous elements together, there exist two types of unions, depending on how flow and effort interact with them.

5.6.1 1

The 1 union consists in a junction where the sum of efforts is zero and the flow that enters and exits is the same:

$$f_1 = f_2 ; \qquad (25)$$

$$e_1 + e_2 = 0 ; \qquad (26)$$

5.6.2 0

The 0 consists in a junction where the sum of flows is zero and the efforts are the same:

$$e_1 = e_2 ; \qquad (27)$$

$$f_1 + f_2 = 0 ; \qquad (28)$$

6 Basic blocks of the model

Once the basic imaginary elements that are used in Bond Graph have been explained, the next step is to describe the different blocks that will allow the creation of a model. These blocks are made to be used in 20Sim, so in order to facilitate the creation of the model sub-blocks with code are included. They are meant to be used in a variety of multi solid models, so their structure is made to be general in order to facilitate the creation of different dynamic systems.

An important characteristic regarding the basic blocks is that they will include two 3D power transfer elements, making them manage 12 different signals. In the following explanations two power paths will be seen, one for the forces and another for the moment of the different solids.

6.1 Translation

The translation block consists in moving the point in space where the flow of energy is located, using a vector, or cross, product. The main use of this block is the translation from the centre of mass of a given rigid solid to a point of interest, where a joint, a friction point or other elements are located [12].

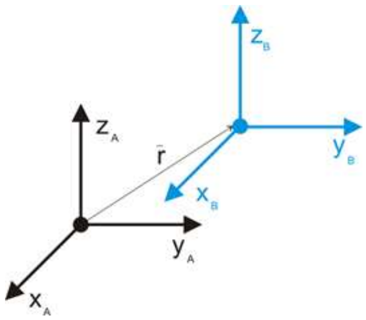


Figure 16. Translation block icon

As previously stated, the block will consist in two cross products, one for each power path. The basic math of the cross product is the following [1]:

$$a \times b = \|a\|\|b\| \sin(\theta)\, n\,; \tag{29}$$

Where a and b are the vectors, θ is the angle between them, ||a|| and ||b|| are the magnitudes and n is the perpendicular unit vector.

In the case of the 20Sim block, one of the vectors will be the distance vector from the original point to the interest one, and the other vector will be the power input. The Bond Graph layout of the translation block is represented in Figure 17.

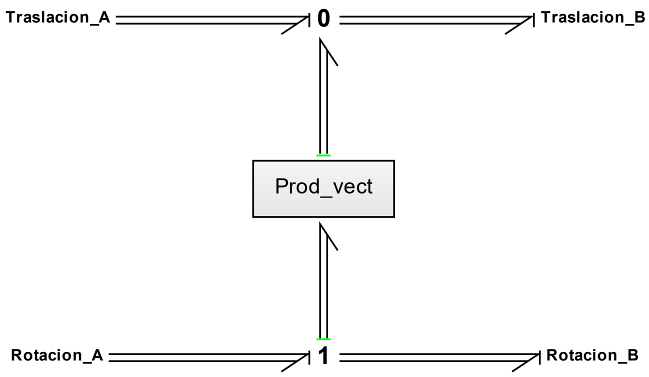


Figure 17. Translation Bond Graph diagram

By observing the block diagram the flow of power can be determined. In the translation case, the input `A` joins a `0` junction, and in the rotation case, which is the input name referring to the angular velocities and momentum (not referring to the rotation block), the flow joins a `1` junction. From these junctions the energy flow is connected to the vector product block. In the case of the forces and velocities (`Translacion_A`), connected with a `0` junction, the vector that will enter the code block (`Prod_vec`) is the Force one (`p.e`). We can observe this thanks to the causality of the power transfer element. In the case of the momentum and angular velocity it is the opposite, a flow will enter. This means that the `0` junction will receive a flow and the `1` junction will receive an effort from the product block.

The moment and angular speed power enters the block, so it is defined `Pin`, with its two components of effort and flow. The force and velocity power exits de block, so it is defined as `Pout`. The code that the block executes is the following:

```
variables
   real global D_VAR_PP_L[3,1];
equations
 Pin.e = cross (-D_VAR_PP_L, Pout.e);
 Pout.f = cross (Pin.f,-D_VAR_PP_L);
```

This code multiplies the force by the distance vector, defined in variables, obtaining a moment that is sent to the `1` junction. Basically it is calculating the moment of the forces applied in the input of the translation block, generally the forces applied to the solid. This will modify the output moment since the sum of all three moments in the junction has to be 0.

In the next line of the code executed within the block, it multiplies the angular velocity to the distance vector, calculating the velocity due to this angular speed at that distance. The resulting velocity is sent to the `0` junction. This will modify the output velocity since in the sum of all three velocities has to be 0.

6.2 Rotation

The rotation block is used to update and modify the reference angles of both of the incoming 3D Bond Graph signals. This blocks main uses will be to set an initial angle between two solids, and to update a given angle when one solid is moving relative to the other [12].

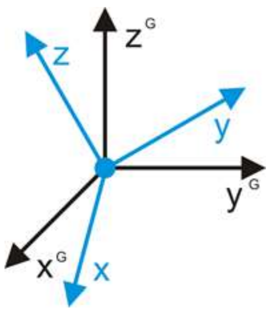


Figure 18. Rotation block icon

The function of the block will be accomplished thanks to a rotation matrix that will update the desired angle of the signal going through. To optimize this, it will usually be set to update only the angles that have a free degree of rotation between solids.

The basic math of a rotation matrix is as follows [1]:

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = R_z(\psi)R_y(\theta)R_x(\phi)\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}; \qquad (30)$$

$$R_z(\psi)R_y(\theta)R_x(\phi) = \begin{Bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{Bmatrix}\begin{Bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{Bmatrix}\begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{Bmatrix}; \qquad (31)$$

This matrix basically rotates a given vector by a desired angle in each of the three axis, and in our model is used to update the yaw, pitch and roll of a solid.

The rotation block diagram in 20Sim is as represented in Figure 19.

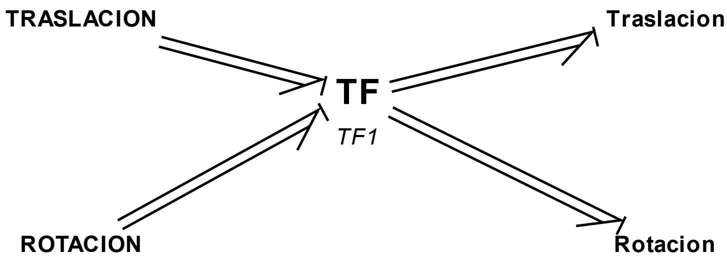


Figure 19. Rotation block diagram

A seen in the diagram, both signals go to a transformation element, but in this case the code has been modified in order to accommodate the rotation matrix and update the angle of rotation. The code of the transformer is as follows:

```
variables
real psi,teta,fi,dpsi,dteta,dfi;
real mpsi[3,3], mteta[3,3],mfi[3,3];

equations
if cos(teta)<>0 then dpsi=[sin(fi)/cos(teta)]*rl.f[2,1]+[cos(fi)/cos(teta)]*rl.f[3,1]; else dpsi=0; end;
dteta=cos(fi)*rl.f[2,1]-sin(fi)*rl.f[3,1];
if cos(teta)<>0 then dfi=rl.f[1,1]+sin(fi)*[sin(teta)/cos(teta)]*rl.f[2,1]+cos(fi)*[sin(teta)/cos(teta)]*rl.f[3,1];
else dfi=0; end;

//psi=int(dpsi);
psi=0;
teta=int(dteta);
//fi=int(dfi);
fi=0;

mfi= [1,0,0;0,cos(fi),-sin(fi);0,sin(fi),cos(fi)];
mteta=  [cos(teta),0,sin(teta);0,1,0;-sin(teta),0,cos(teta)];
mpsi=  [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];

lg.f=mpsi*mteta*mfi*ll.f;                        //global lineal velocity
rg.f=mpsi*mteta*mfi*rl.f;                       //global angular velocity
ll.e=transpose(mpsi*mteta*mfi)*lg.e;      //local linear force
rl.e=transpose(mpsi*mteta*mfi)*rg.e;     //local momentum
```

There are two main parts of this code. The first equations found are used to update the angles between the input and output signals. Then the variation in the angle is assigned to the angles that will rotate in the model. And finally the rotation matrix appears, which will modify the signal depending on the recorded angles on the first part.

6.3 Rigid Solid

The rigid solid block will, as its name implies, define a solid of the model [12]. Each of the solids will have this block assigned to them, where its centre of mass, inertia and mass will be set.

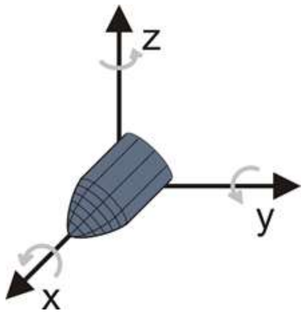


Figure 20. Rigid Solid icon

As previously said, the function of this block is to define the different parameters of a given solid. This includes setting the centre of mass, assigning a mass and the vertical weight, and setting its inertia.

Before moving to the 20Sim block, the equations of the movement of a solid used have to be gone through. The used equations are the Newton-Euler ones and the block is mainly based around them.

They are obtained by applying the conservation of momentum and angular momentum to a body. The Euler's equation of conservation of linear momentum is applied [12]:

$$\sum \vec{f_i} = \frac{d\vec{p}}{dt} + \vec{w} \times \vec{p} = m\vec{a} + \vec{w} \times m\vec{v} \,; \qquad (32)$$

Where f are the forces, m is the mass, v the velocity and a the acceleration of the body.

And the Euler equation to describe the conservation of angular momentum of a rigid body [12]:

$$\sum \vec{t_i} = \frac{d\vec{h}}{dt} = \boldsymbol{I}\,\vec{\propto} + \vec{w} \times \vec{h} \,; \qquad (33)$$

Where I is the inertia, w the angular velocity, ∝ the angular acceleration and h the kinetic movement.

Finally the Newton-Euler equations expressed in matrix form are the following:

$$\begin{Bmatrix} \vec{f} \\ \vec{t} \end{Bmatrix} = \begin{Bmatrix} m & 0 \\ 0 & \boldsymbol{I} \end{Bmatrix} \begin{Bmatrix} \vec{a} \\ \vec{\propto} \end{Bmatrix} + \begin{Bmatrix} \vec{w} \times m\vec{v} \\ \vec{w} \times \boldsymbol{I}\vec{w} \end{Bmatrix} ; \qquad (34)$$

This equation defines the forces and moments applied to a solid due to an acceleration or a rotation. This equations are useful because they can be coded with the effort and flow parameters:

$$\begin{Bmatrix} \overrightarrow{p.e_t} \\ \overrightarrow{p.e_r} \end{Bmatrix} = \begin{Bmatrix} m & 0 \\ 0 & \boldsymbol{I} \end{Bmatrix} \begin{Bmatrix} \overrightarrow{p.\dot{f_t}} \\ \overrightarrow{p.\dot{f_r}} \end{Bmatrix} + \begin{Bmatrix} \overrightarrow{p.f_r} \times m\overrightarrow{p.f_t} \\ \overrightarrow{p.f_r} \times \boldsymbol{I}\overrightarrow{p.f_r} \end{Bmatrix} ; \qquad (35)$$

To represent these equations plus the gravity effect the used diagram in 20Sim is illustrated in Figure 21.

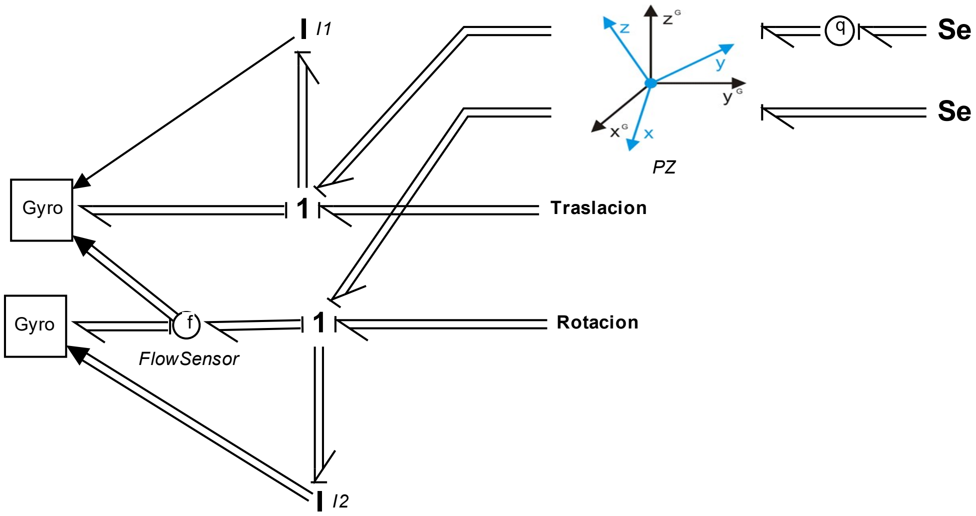


Figure 21. Rigid Solid block diagram

Firstly the `1` junctions represent the point where all the forces and momentums that affect the body will be added. Through the inputs `Translation` and `Rotation` the external ones will enter the block, and within the block the ones created due to the mass and inertia of the solid will be applied.

The gravity is added with the top source of effort (`Se`), which enters a rotation block in order to make the weight of the solid always point in the vertical direction. It also tracks with a `qsensor` the origin of this weight, the centre of mass, which is useful for obtaining the coordinates of the position of the solid during the simulation, and are also useful for the animation of the model. In case the solid has an applied moment, the second effort source can be used.

The rest of the block is dedicated to solving the Newton-Euler equations. The two inertia blocks are tasked by solving the following part of the equations:

$$\begin{Bmatrix} p.e_t \\ p.e_r \end{Bmatrix} = \begin{Bmatrix} m & 0 \\ 0 & I \end{Bmatrix} \begin{Bmatrix} p.\dot{f}_t \\ p.\dot{f}_r \end{Bmatrix} + \begin{Bmatrix} p.f_r \times mp.f_t \\ p.f_r \times Ip.f_r \end{Bmatrix};$$

It operates as described previously, but in the rotation side of the equation the mass is substituted by the inertia. The code used inside is the following:

```
parameters
   real global masa_V1;
equations
    state = int(p.e);
    p.f = state / masa_V1;
    m=masa_V1;
```

```
parameters
   real global inercia_V1 [3,1];
equations
    state = int(p.e);
    p.f = state ./ inercia_V1;
    J=inercia_V1;
```

Finally the `Gyro` blocks are used to calculate the rest of the Newton-Euler equations:

$$\begin{Bmatrix} p.e_t \\ p.e_r \end{Bmatrix} = \begin{Bmatrix} m & 0 \\ 0 & I \end{Bmatrix} \begin{Bmatrix} p.\dot{f}_t \\ p.\dot{f}_r \end{Bmatrix} + \begin{Bmatrix} p.f_r \times mp.f_t \\ p.f_r \times Ip.f_r \end{Bmatrix};$$

For the conservation of linear momentum, calculated in the top `Gyro`, the signal of the angular velocity of the solid is imported by a flow sensor and the value of the mass is imported form the top inertia block. The code inside the `Gyro` is the following:

```
equations
    P.e = cross (W, m* P.f);
```

For the component of the Newton-Euler equation for the conservation of angular momentum, a `Gyro` is also used and the value of inertia is imported from the bottom inertia block. The code inside the `Gyro` is the following:

```
equations
   P.e = cross (P.f, J .* P.f);
```

By looking at the causality it can be seen that all the moments and forces are added in the `1` junction, and the resulting moment or force is sent to the `I` block to give a speed and an angular speed to the solid. This is how the movement of a given solid using this block is obtained.

6.4 Joint

The joint block is used to restrict the movement between two solids in one axis of rotation [12]. The main use of this block is to constrain the movement between two solids, making a union, and to leave one axis of rotation free, but it can be easily modifiable to include more free axis of rotation.

Figure 22. Joint block icon

The basic function of this block is to assign a big force or moment to the movements and axis of rotation that have to be constrained. This is done with a `C` element, giving a really high value of the `k` constant. The 20Sim Bond Graph diagram represented in Figure 23.

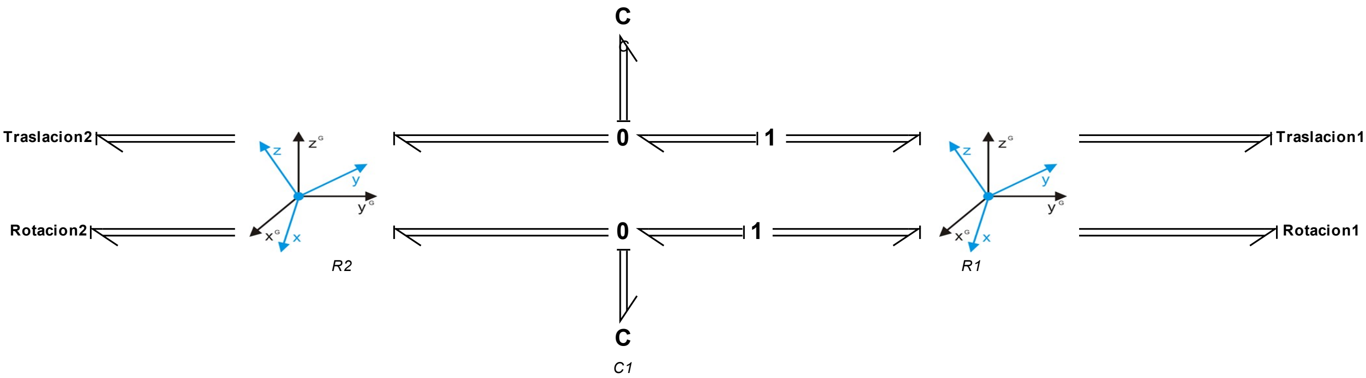


Figure 23. Joint block diagram

The flow constraint is set in the `0` junction with the `C` element, and the `1` junctions are used for having the correct power flow direction when entering the rotation blocks. These blocks are used to set an initial angle between the joined elements, and to update the angle between them.

Even though it has been previously stated that a `k` constant with a high value is used, to improve the simulation calculations a constraint function for the flow is used, aiding in reducing the simulation time. The code of the translation `C` element, the top one, is the following:

```
equations
p.e = constraint (p.f);
//p.e = 1e10 * int(p.f);
// p.e = k * int(p.f);
```

And the code for the rotation C element, the bottom one, is the following:

```
variables
 real a, b;
equations
a = constraint(p.f[1,1]);
b = constraint(p.f[2,1]);
//c = constraint(p.f[3,1]);
    //p.e[1,1] = 1e8 * int(p.f[1,1]);
    //p.e[2,1] = 1e8 * int(p.f[2,1]);
    p.e[1,1] = a;
    p.e[2,1] = b;
    p.e[3,1] = 0;
```

As seen in this code, the free axis of rotation is assigned a moment with value 0, and the other axis are constrained, meaning that the movement is restricted. If a constant was used, the joint would behave like a spring with a really high spring rate, which would increase the calculation times since there would exist small and periodic movements in the joint.

6.5 Prismatic Joint

The prismatic joint block is a variation of the joint block, and this variation is that instead of leaving a free axis of rotation, a free movement axis is enabled with all the axis off rotation being constrained [12][14]. This block is used when the movement in one axis between two objects is needed. One good example would be the union between the piston rod and the piston tube.

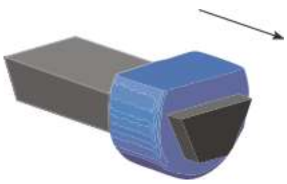


Figure 24. Prismatic joint icon

The 20Sim diagram of a prismatic joint is represented in Figure 25.

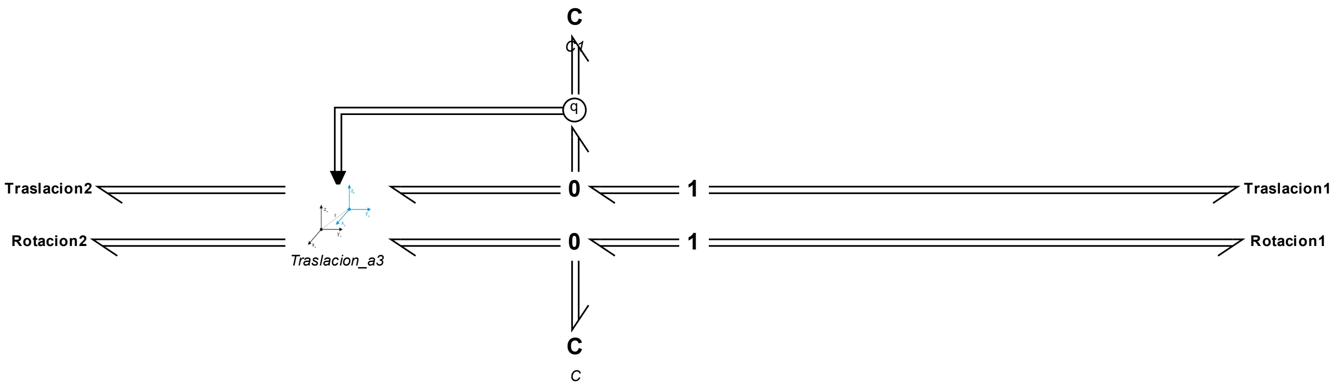


Figure 25. Prismatic Joint block diagram

Since the rotations are constrained, there is no need to track and update the relative angle between both solids. It will be explained forward in the report, but when this joint is used is usually accompanied by two normal joints at each end to fulfil the function of giving a free axis of rotation. Since two solids will be moving relative to each other there is the need of tracking and updating the distance between them. This is done by recording the speed of the free axis and sending it as a signal to a modified translation. The modified translation block is represented in Figure 26.

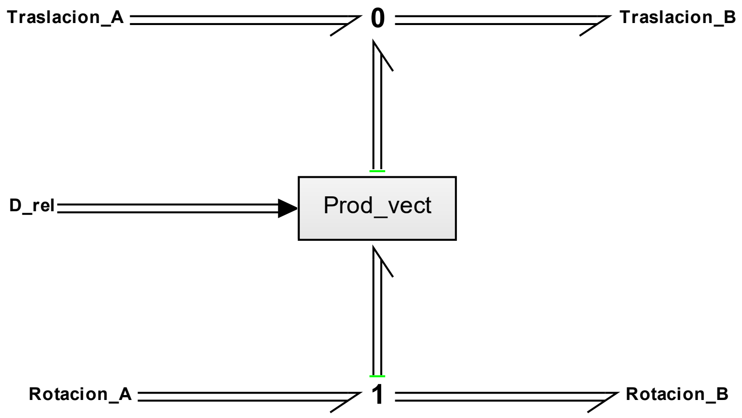


Figure 26. Modified translation block diagram

The basic modification is that the speed flow is recorded and integrated, sending the relative distance between the two elements connected to the joint. This relative distance is cross multiplied with the effort and flow of the rotation and translation power transmissions as explained in the translation block part.

As seen previously, this is the code inside the `Prod_vect` block:

```
equations
 Pin.e = cross (-d, Pout.e);
 Pout.f = cross (Pin.f, -d);
```

The value of `d` will be continuously updated through the simulation, depending on the relative movement between the elements connected to the prismatic joint. The way it functions is that it integrates the velocity in the free axis and assigns the value to the `d` variable.

7 Multi solid model

Once the basic building blocks have been described, the next step of the project is to assemble the model. To do so the different blocks will be arranged and joined to form a set of sub systems that, when assembled, will form the model. Before creating any sub system, the first step is to define all the parameters that have been obtained in the vehicle description part. Once the definition has been made, the next step is to tackle the different assemblies of blocks to create the sub systems, which are: the piston assembly, the central articulation, the rear wheel assembly, the front steering assembly, the tyre contact system and the hydraulic system. To finalise the model a set of plots and an animation have been made in order to record and present the results of the simulation.

7.1 Parameter definition

The parameter definition consists in creating an isolated block in the model where all the geometric and dynamic parameters are defined or obtained. This feature is not necessary, the different parameters can be defined within each basic block, but since the aim of the project is to have a model that can easily be modified to simulate different vehicles, it is included.

The first step of the parameter definition is to include in the block all the geometric points and mass and inertia values which have been obtained in part 4 of the project. This is done firstly by ordering the masses, inertia and centre of mass of every solid, and then defining all the interest points:

```
//Globals
parameters
 real global
//V1
 masa_V1 = 1210,
 inercia_V1[3,1] = [127.23;338.93;291.75],
 CM_V1[3,1] = [0.309;0;0.5996],

//V2
 masa_V2 = 650,
 inercia_V2[3,1] = [41.62;87.12;62.83],
 CM_V2[3,1] = [-1.4472;0;0.4715],

//VBR
 masa_VBR = 80,
 inercia_VBR[3,1] = [5.02;8.84;5.02],
 CM_VBR[3,1] = [-1.495;-0.575;0.423],

//VBL
 masa_VBL = 80,
 inercia_VBL[3,1] = [5.02;8.84;5.02],
 CM_VBL[3,1] = [-1.495;0.575;0.423],

//VFR
 masa_VFR = 90,
 inercia_VFR[3,1] = [5.02;8.84;5.02],
 CM_VFR[3,1] = [0;-0.575;0.423],
```

```
//VFL
 masa_VFL = 90,
 inercia_VFL[3,1] = [5.02;8.84;5.02],
 CM_VFL[3,1] = [0;0.575;0.423],

//VAR
 masa_VAR = 10,
 inercia_VAR[3,1] = [1;1;1],
 CM_VAR[3,1] = [0;-0.575;0.423],

//VAL
 masa_VAL = 10,
 inercia_VAL[3,1] = [1;1;1],
 CM_VAL[3,1] = [0;0.575;0.423],

//Pistons
 masa_P = 10,
 inercia_P[3,1] = [1;1;1],
//P1
 CM_P1A[3,1] = [-0.305;-0.105;0.573],
 CM_P1B[3,1] = [-0.658;-0.105;0.573],
 R_P1[3,1] = [-0.4815;-0.105;0.573],
//P2
 CM_P2A[3,1] = [-0.305;0.105;0.573],
 CM_P2B[3,1] = [-0.658;0.105;0.573],
 R_P2[3,1] = [-0.4815;0.105;0.573],
```

```
//P3
 CM_P3A[3,1] = [0.0625;-0.22;0.573],
 CM_P3B[3,1] = [0.0625;-0.22;0.573],
 R_P3[3,1] = [0.0625;-0.22;0.573],
//P4
 CM_P4A[3,1] = [0.0625;0.22;0.573],
 CM_P4B[3,1] = [0.0625;0.22;0.573],
 R_P4[3,1] = [0.0625;0.22;0.573],
//BarraDirección
 BD_1[3,1] = [-0.141945;0.329;0.573],
 BD_2[3,1] = [-0.141945;0;0.573],
 BD_3[3,1] = [-0.141945;-0.329;0.573],

//PR
 P_R12[3,1] = [-0.65;0;0.573],
 P_R1FR[3,1] = [0;-0.375;0.423],
 P_R1FL[3,1] = [0;0.375;0.423],
 P_R2BR[3,1] = [-1.495;-0.575;0.423],
 P_R2BL[3,1] = [-1.495;0.575;0.423],
 P_TFR[3,1] = [0;-0.575;0],
 P_TFL[3,1] = [0;0.575;0],
 P_TBR[3,1] = [-1.495;-0.575;0],
 P_TBL[3,1] = [-1.495;0.575;0],
 P_1AR[3,1] = [0;-0.375;0.423],
 P_1AL[3,1] = [0;0.375;0.423],
//PP
 PP_S[3,1] = [-0.658;0.105;0.573],
 PP_P[3,1] = [-0.658;-0.105;0.573],
 PP_R[3,1] = [-0.305;0.105;0.573],
 PP_Q[3,1] = [-0.305;-0.105;0.573],
 PP_J[3,1] = [0.04;0.05;0.573],
 PP_K[3,1] = [0.04;-0.05;0.573],
 PP_I[3,1] = [0.085;0.39;0.573],
 PP_L[3,1] = [0.085;-0.39;0.573],

//Psi_ini
 PsiP3_ini = 97.54,
 PsiP4_ini = 82.46,
 k = 1000000000;
```

As it can be seen at the end of the parameter definition, two angles are defined in order to give an initial value to the joints where the pistons will be mounted. The next step of the parameter definition is to create the different vectors and values that are also necessary for the blocks in the model, especially in the translations. In their case, these vectors are the distance between the different centres of masses and the points where a joint is located. Also the weights labelled as `peso` are defined:

```
variables
 real global
 peso_V1[3,1], peso_V2[3,1], peso_VBR[3,1],peso_VBL[3,1],peso_VFR[3,1],
 peso_VFL[3,1],peso_VAR[3,1],peso_VAL[3,1],
 D_V1_P_R12[3,1],D_V2_P_R12[3,1],
 D_V1_P_R1FR[3,1],D_FR_P_R1FR[3,1],
 D_V1_P_R1FL[3,1],D_FL_P_R1FL[3,1],
 D_V2_P_R2BR[3,1],D_BR_P_R2BR[3,1],
 D_V2_P_R2BL[3,1],D_BL_P_R2BL[3,1],
 D_FR_P_TFR[3,1],D_FL_P_TFL[3,1],D_BR_P_TBR[3,1],D_BL_P_TBL[3,1],
 D_V2_PP_S[3,1],D_V2_PP_P[3,1],
 D_V1_PP_R[3,1],D_V1_PP_Q[3,1],D_V1_PP_J[3,1],D_V1_PP_K[3,1],
 D_VAR_P_1AR[3,1],D_VAR_PP_L[3,1],
 D_VAL_P_1AL[3,1],D_VAL_PP_I[3,1],
 D_V1_P_1AR[3,1],D_V1_P_1AL[3,1],
 D_P1A_RP1[3,1],D_P1B_RP1[3,1],D_P2A_RP2[3,1],D_P2B_RP2[3,1],D_P3A_RP3[3,1],
 D_P3B_RP3[3,1],D_P4A_RP4[3,1],D_P4B_RP4[3,1],
 psi_ini_P3_rad, psi_ini_P4_rad,
 D_FR_BD3[3,1],D_FL_BD1[3,1],D_BD1_BD2[3,1],D_BD3_BD2[3,1];
```

```
equations
peso_V1 = [0;0;-masa_V1*g_n];
peso_V2 = [0;0;-masa_V2*g_n];
peso_VBR = [0;0;-masa_VBR*g_n];
peso_VBL = [0;0;-masa_VBL*g_n];
peso_VFR = [0;0;-masa_VFR*g_n];
peso_VFL = [0;0;-masa_VFL*g_n];
peso_VAR = [0;0;-masa_VAR*g_n];
peso_VAL = [0;0;-masa_VAL*g_n];
D_V1_P_R12 = P_R12 - CM_V1;
D_V2_P_R12 = P_R12 - CM_V2;
D_V1_P_R1FR = P_R1FR - CM_V1;
D_FR_P_R1FR = P_R1FR - CM_VFR;
D_V1_P_R1FL = P_R1FL - CM_V1;
D_FL_P_R1FL = P_R1FL - CM_VFL;
D_V2_P_R2BR = P_R2BR - CM_V2;
D_BR_P_R2BR = P_R2BR - CM_VBR;
D_V2_P_R2BL = P_R2BL - CM_V2;
D_BL_P_R2BL = P_R2BL - CM_VBL;
D_FR_P_TFR = P_TFR - CM_VFR;
D_FL_P_TFL = P_TFL - CM_VFL;
D_BR_P_TBR = P_TBR - CM_VBR;
D_BL_P_TBL = P_TBL - CM_VBL;
D_V2_PP_S = PP_S - CM_V2;
D_V2_PP_P = PP_P - CM_V2;
D_V1_PP_R = PP_R - CM_V1;
D_V1_PP_Q = PP_Q - CM_V1;
D_V1_PP_J = PP_J - CM_V1;
D_V1_PP_K = PP_K - CM_V1;
D_VAR_P_1AR = P_1AR - CM_VAR;
D_VAR_PP_L = PP_L - CM_VAR;
D_VAL_P_1AL = P_1AL - CM_VAL;
D_VAL_PP_I = PP_I - CM_VAL;
D_V1_P_1AR = P_1AR - CM_V1;
D_V1_P_1AL = P_1AL - CM_V1;
D_P1A_RP1 = R_P1 - CM_P1A;
D_P1B_RP1 = R_P1 - CM_P1B;
D_P2A_RP2 = R_P2 - CM_P2A;
D_P2B_RP2 = R_P2 - CM_P2B;
D_P3A_RP3 = R_P3 - CM_P3A;
D_P3B_RP3 = R_P3 - CM_P3B;
D_P4A_RP4 = R_P4 - CM_P4A;
D_P4B_RP4 = R_P4 - CM_P4B;
psi_ini_P3_rad = PsiP3_ini*(3.14/180);
psi_ini_P4_rad = PsiP4_ini*(3.14/180);
D_FR_BD3 = BD_3 - CM_VAR;
D_FL_BD1 = BD_1 - CM_VAL;
D_BD1_BD2 = BD_1 - BD_2;
D_BD3_BD2 = BD_3 - BD_2;
```

All of these distances will be used in the translation blocks, and the weights (`pesos`) in the rigid body blocks as explained in their respective parts. The importance of this block is that by changing the initial parameters a user can change the whole geometry, inertia and weights of the vehicle. By looking at the variables defined it can be seen that the weights of the pistons are not there. Since they are the same as the articulation one (10 kg), the weight of the articulations has been assigned to them.

For future use, this block can be easily modified to adapt to the nomenclature of the points that any user wants to set.

7.2 Piston assembly

The piston assembly is one of the main sub systems of the model, which will be the bridge between the dynamic model and the different hydraulic systems to be tested. The description of this sub assembly aims to represent the Bond Graph assembly of a piston, including the mechanism to actuate the dynamic system with the hydraulic one.

The first step is to design a new block that includes the multi solid assembly of the piston, with the aim of being able to duplicate it in an easy way, without having to significantly modify the parameters that define it. The block assembly of the piston is represented in Figure 27.

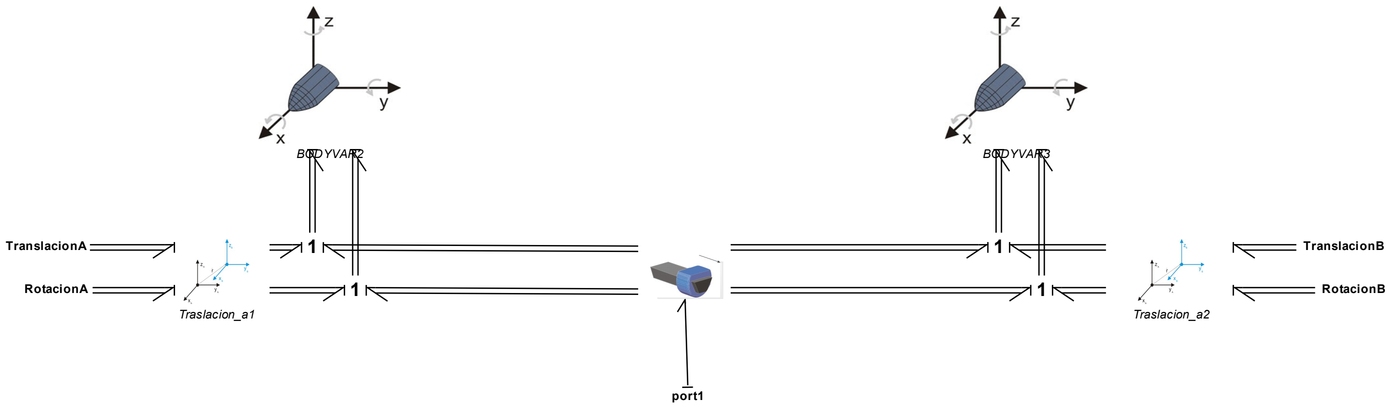


Figure 27. Piston Bond Graph assembly

As it can be seen in the assembly, there are the two rigid bodies defining the barrel and the rod of the piston. By looking at this graph, the position of the centre of masses of both the piston parts will be initially in the same position, since there are no translations between the rigid solids and the joint. Before the connections exiting the block there is a translation that moves the power transmission from the centre of the piston to the mounting point, where there will probably be a joint connected to it, but it is preferred to leave it outside the piston block. The vector inside the translation is obviously the distance from the joint to the union points of the piston.

The power signal will enter the piston block, will go through a translation from the mounting point to the joint of the piston. There will be a `1` junction, where the sum of efforts will be done and sent to the rigid body block in order to calculate the dynamics of the barrel / rod. Another signal will come to the `1` junction with the efforts due to the restrictions of the joint, which in turn will affect the flow obtained inside the body block.

As it can be seen, there is also a port entering the prismatic joint, which is not something that has been described in its explanation. It has been modified in order to incorporate the hydraulic Bond Graph. The modified prismatic joint diagram is represented in Figure 28.

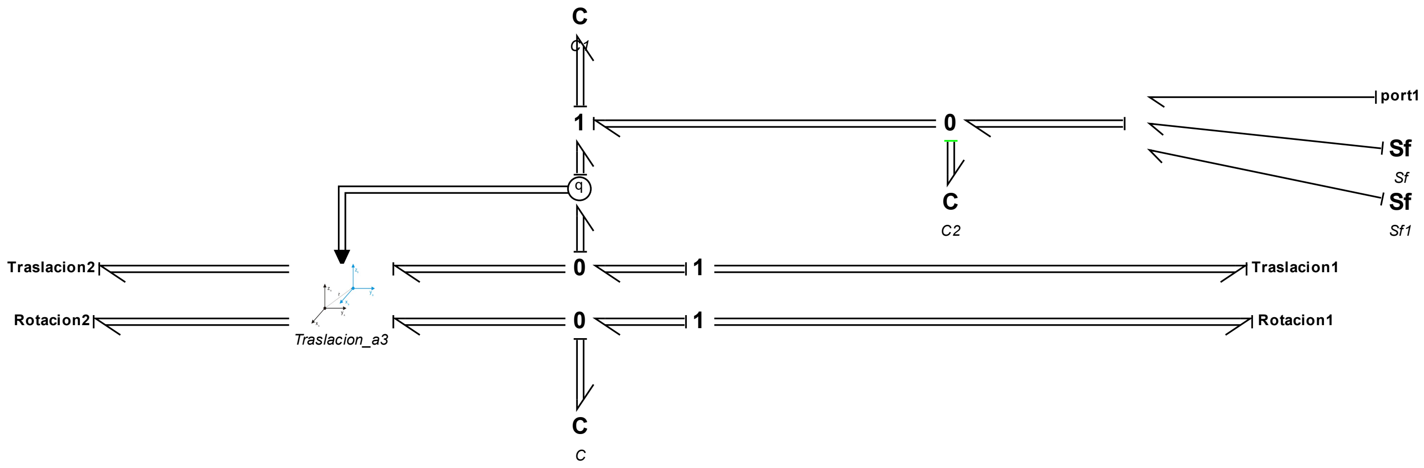


Figure 28. Modified prismatic joint diagram

A new diagram has been added to the joint. The aim of this diagram is to set a velocity coming from the hydraulic Bond Graph. By looking at the causality the flows and efforts can be followed to determine how the flow coming from the hydraulic bond graph is transformed into an effort that actuates de pistons to make them move. The flow comes from the hydraulic system, which is done with 1D signals. This signal is added with a mux to two source flows with a value of 0, resulting in a 3D vector.

This vector enters a `0` junction, and by the causality it can be seen that a flow is entering this junction. Since it is a `0` union, the sum of flows will have to be zero. By looking at the causality, the flow will go to the Capacitor, which will have the movement in the pistons direction, x, constrained.

```
variables
 real a;
equations
a = constraint(p.f[1,1]);
//b = constraint(p.f[2,1]);
//c = constraint(p.f[3,1]);
//p.e[1,1] = 1e8 * int(p.f[1,1]);
//p.e[2,1] = 1e8 * int(p.f[2,1]);
    p.e[1,1] = a;
    p.e[2,1] = 0;
    p.e[3,1] = 0;
```

This will return an effort to the `0` junction, and by its properties we can see that this effort will go in both directions, going to the prismatic junction and returning to the hydraulic system. This is because in the `0` unions the effort is equal for all entering and exiting power arrows. Since the movement is constrained, no flow will go to the Capacitor, so the flow entering from the hydraulic system will be the same as the one sent to the prismatic joint. With this diagram it is possible to calculate the effort required to move the pistons by assigning a flow coming from the hydraulic system. It also enables the testing of the multi solid model by assigning a theoretical velocity of the pistons with a flow source.

Another advantage of compressing the piston assembly in a block is the organization of the overall model. Since it will be necessary to have the overall geometry of the model done without compressing it into sub-assemblies, to have a better overall idea of how it works, organizing these sub-assemblies into blocks will avoid the model to be crowded and hard to interpret.

7.3 Central articulation

The next step in creating the model is to make the assembly of the core or the vehicle. This core is composed of the two main volumes of the tractor, the front and back parts. These parts make up the majority of the weight, and are the points were all of the wheels and steering components will be mounted to. This sub-assembly represents two rigid solids attached by a joint. The initial, and most basic, assembly of the central articulation is as represented in Figure 29.

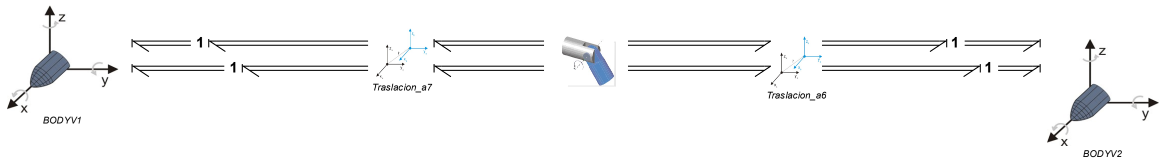


Figure 29. Central articulation assembly

It can be seen in the Bond Graph diagram that a translation will send the efforts that result from the constraints in the joints to the rigid solid blocks, affecting their movement and behaviour. The free axis of rotation will be `psi`, the axis in the z direction. To do so the two other axis of rotation will be constrained, and the rotations inside the joint will have to update this angle.

The distance vectors defined in the parameters block for this distances are the following:

```
D_V1_P_R12 = P_R12 - CM_V1;
D_V2_P_R12 = P_R12 - CM_V2;
```

Where `P_R12` is the central joint, or `C` in the drawing of the vehicle description.

The joint structure is the generic one explained previously:

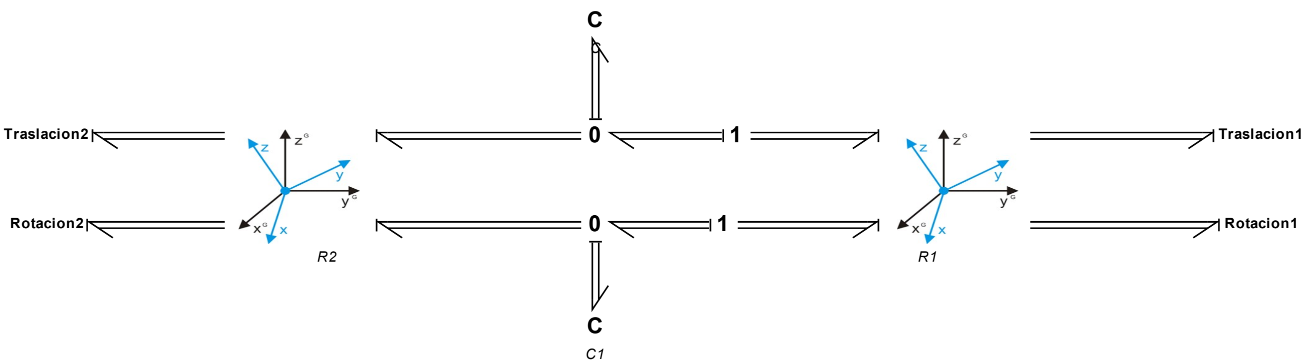


Figure 30. Joint diagram

Code inside the translation capacitor:

```
equations
p.e = constraint (p.f);
// p.e = 1e12 * int(p.f);
// p.e = k * int(p.f);
```

Code inside the rotation capacitor:

```
variables
 real a, b;
equations
a = constraint(p.f[1,1]);
b = constraint(p.f[2,1]);
//c = constraint(p.f[3,1]);
//    p.e[1,1] = 1e15 * int(p.f[1,1]);
//    p.e[2,1] = 1e15 * int(p.f[2,1]);
    p.e[1,1] = a;
    p.e[2,1] = b;
    p.e[3,1] = 0;
```

In the rotation matrices the code is set to only update `psi`, and the rest is the same as in the generic rotation matrix shown in its part:

```
psi=int(dpsi);
//teta=int(dteta);
//fi=int(dfi);
teta=0;
fi=0;
```

Even though this sub-assembly explains the main mechanism, the actual central articulation also includes a pair of pistons mounted to both of the two parts of the tractor. The final sub-assembly of the central articulation is represented in Figure 31.

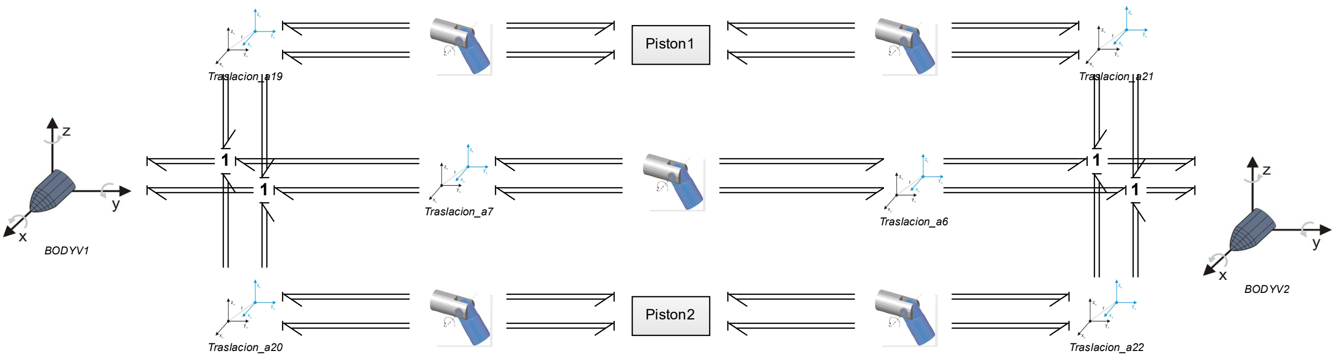


Figure 31. Central articulation full assembly diagram

As seen in the Bond Graph diagram, both pistons are added. This is done by creating a translation from the centre of mass of the main volumes to the piston mounting points, and putting a joint to articulate the piston with one free rotation axis. The distance vectors of the translation are the following:

For piston 1:

```
D_V1_PP_Q = PP_Q - CM_V1;

D_V2_PP_P = PP_P - CM_V2;
```

For piston 2:

```
D_V1_PP_R = PP_R - CM_V1;

D_V2_PP_S = PP_S - CM_V2;
```

And the joints are the same as the central one, giving a free rotation in the z axis while constraining the others, and the movement. These joints emulate a piston mounted with a bolt to the two main chassis of the tractor. If more free axis of rotation are needed, the extra axis in the capacitor has to be set to 0 and the angle associated has to be updated in the rotation block. To do so, simply uncommenting the code in green for the selected angle, and erasing the one that sets it to 0, would be enough. This can be seen in the code in the previous page where the angle is set.

In this diagram there are two missing Bond Graph power signals going to the pistons from the hydraulic system.

7.4 Rear wheel assembly

The rear wheel sub-assembly is one of the simplest in the model. It consists in the direct union of the wheel to the back chassis, since in the description part it was decided that the axels where solid mounted to the main frames of the tractor. This reduces the complexity of the assembly since the centre of mass of the wheel and the union point can be the same, and on top of that this joining is done directly to the `V2` rigid solid.

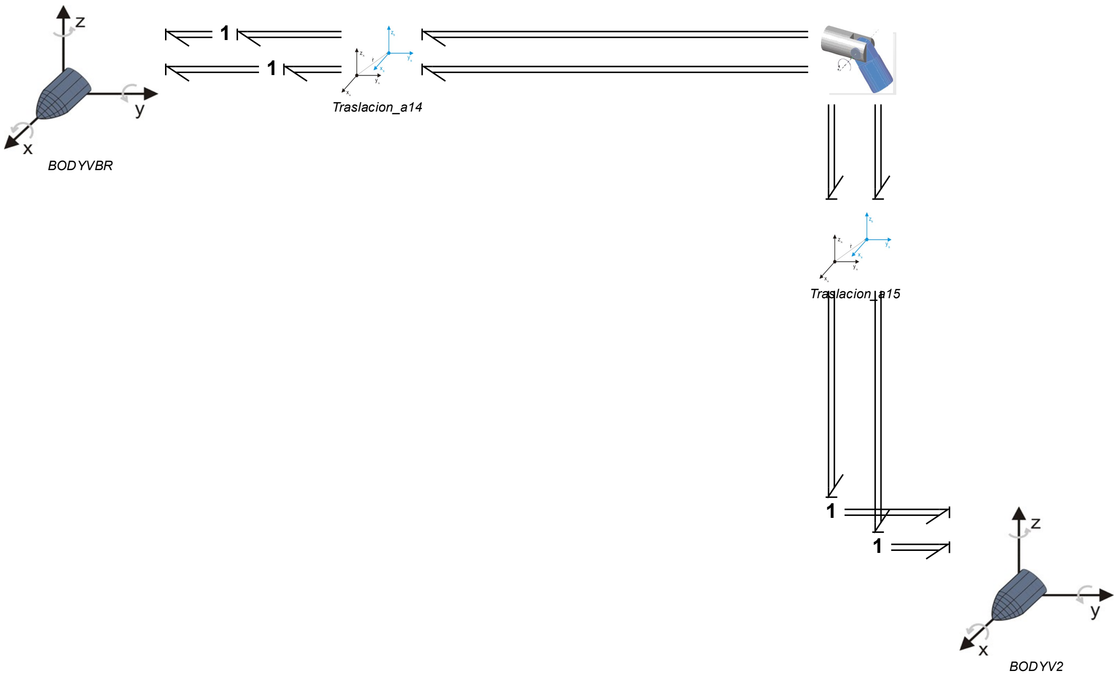


Figure 32. Rear wheel assembly diagram

The structure is basically the same one as in the central articulation, the first one and the most basic. But in this case the free axis of rotation is a different one since it is going to be enabling the wheel to turn freely. This axis is the one in the y direction, represented by the angle `theta`. It is basically the axis of rotation of the wheel.

The distance vectors for the translations are the following:

```
D_V2_P_R2BR = P_R2BR - CM_V2;
D_BR_P_R2BR = P_R2BR - CM_VBR;
```

The rotation capacitor of the joint is the following:

```
parameters
   real global k;
variables
 real a, c;
equations
a = constraint(p.f[1,1]);
//b = constraint(p.f[2,1]);
c = constraint(p.f[3,1]);
    p.e[1,1] = a;
    p.e[2,1] = 0;
    p.e[3,1] = c;
```

And the rotation matrix to update the `theta` angle of rotation, the one in the y axis:

```
//psi=int(dpsi);
psi=0;
teta=int(dteta);
//fi=int(dfi);
fi=0;
```

This sub assembly is one of the most basic, but it is not the whole rear wheel sub assembly. The missing blocks and diagrams are the ones designed to emulate the tyre contact with the ground, and they will be explained after the front steering mechanism. If the modelled vehicle was rear wheel drive, in the rotation `1` junction of the wheel, the bottom one, an effort source could be added to emulate the torque of the engine or the output of the transmission/ hydraulic motors. This can be done by giving a fixed value or choosing a modulated source to vary the input to simulate real acceleration. The same can be done for the brakes, in this case applying a negative torque that emulates them.

Obviously, this assembly is mirrored for the back left wheel.

7.5 Front steering

The front steering is the most complex sub-assembly of the model. As seen in the drawings, it consists in two steering hubs and two pistons assembled to the main frame and a bar connecting both articulated hubs [15]. To these hubs, the wheels, the steering bar and the pistons are joined to complete the front steering system. This system makes the wheels dependent to each other so a correct turning of the vehicle can be achieved. Without the steering bar both wheels would be independently controlled by the pistons, demanding a more precise and accurate hydraulic system to ensure a proper turning. Another major advantage of this type of steering is in case of a piston breaking. In this case without the steering bar the vehicle would be unusable.

The front steering assembly has been divided in three sub-assemblies in order to easily explain the diagrams. The first sub assembly will be the union of the main frame, the steering hub and the wheel. Secondly the mounting of the piston to the steering hub and thirdly the mounting of the steering bar to the two hubs.

7.5.1 Front steering hub and wheel assembly

The front steering hub and wheel sub assembly consists in the union between the hubs and the wheels, and between the hubs and the main chassis. As previously mentioned, the centre of mass of the hubs are the same as the wheels, reducing the blocks needed to recreate this sub-assembly. The block diagram of the steering hub and wheel assembly is represented in Figure 33.

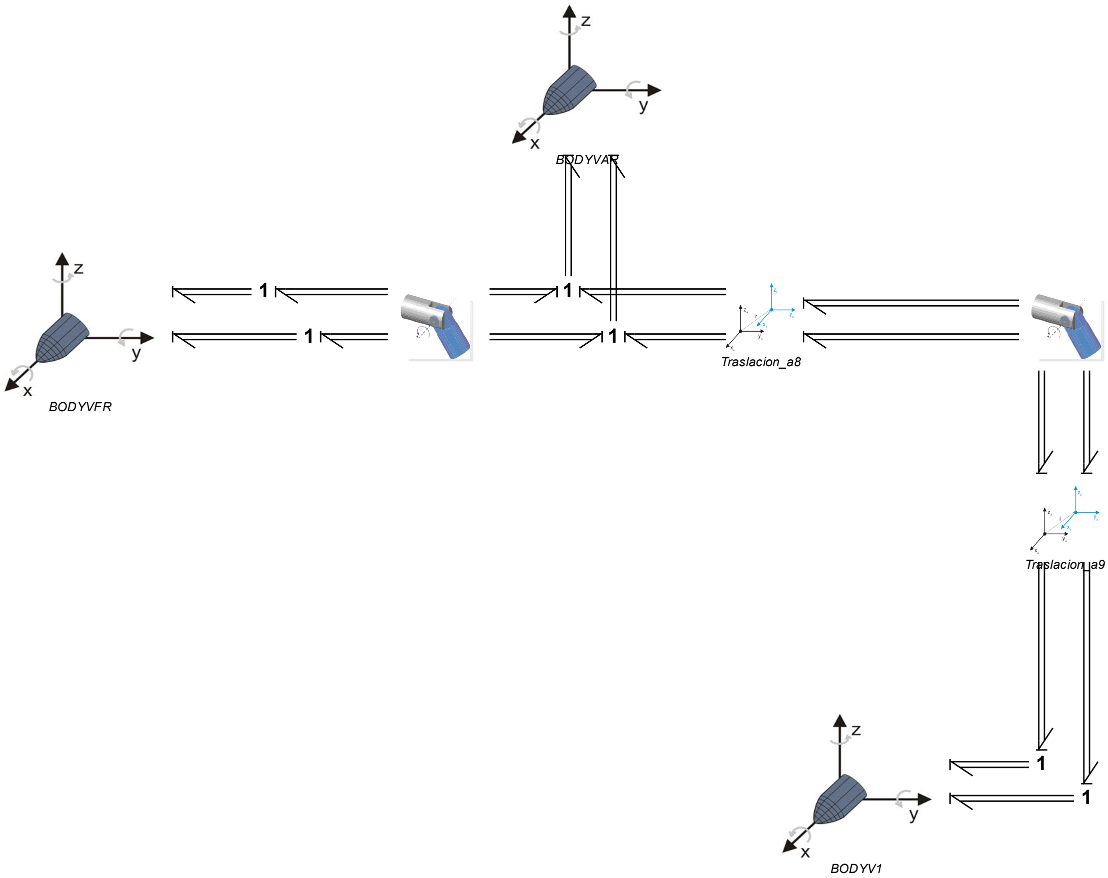


Figure 33. Front steering hub and wheel assembly diagram

As seen in the Bond Graph block diagram, a similar structure to the rear wheel assembly is followed to join the hub to the chassis. A translation is used to move the power transfer from the centre of mass of the main frame to the joint between it and the hub. The vector used in this translation is the following:

```
D_V1_P_1AR = P_1AR - CM_V1;
```

The joint leaves the z axis of rotation free. The translation capacitor inside the joint is the same as all the normal joints, it constraints all the movements:

```
parameters
 real global k;
equations
p.e = constraint (p.f);
```

And in the case of the rotation capacitor, the `psi` angle of rotation is left free:

```
parameters
 real global k;

variables
real a, b;
equations
a = constraint(p.f[1,1]);
b = constraint(p.f[2,1]);
//c = constraint(p.f[3,1]);
//    p.e[1,1] = 1e9 * int(p.f[1,1]);
//    p.e[2,1] = 1e9 * int(p.f[2,1]);
    p.e[1,1] = a;
    p.e[2,1] = b;
    p.e[3,1] = 0;
```

Since the z rotation axis is set free, the rotations inside the joint have to update the angle `psi`:

```
psi=int(dpsi);
//teta=int(dteta);
//fi=int(dfi);
teta=0;
fi=0;
```

An to finalise the union between the hub (`VAR`) and the chassis a translation from the centre of mass to the union point is made, using the distance vector from the centre of mass of the articulated hub to the joint:

```
D_VAR_P_1AR = P_1AR - CM_VAR;
```

As previously mentioned, the centres of masses of the articulated hub and the wheel are set at the same point, so no translations are needed to join the two parts together. The joint to unite both parts has to leave a free axis of rotation to enable the wheel to turn. This axis is the one in the y direction:

```
parameters
   real global k;
variables
 real a, c;
equations
a = constraint(p.f[1,1]);
//b = constraint(p.f[2,1]);
c = constraint(p.f[3,1]);
    p.e[1,1] = a;
    p.e[2,1] = 0;
    p.e[3,1] = c;
```

The rotations in the joint also have to update the `theta` angle:

```
//psi=int(dpsi);
psi=0;
teta=int(dteta);
//fi=int(dfi);
fi=0;
```

7.5.2 Front steering piston assembly

The steering piston sub-assembly is done similarly to the ones in the central articulation. It represents the assembly of a piston, of which the barrels side is mounted to the chassis and the rods side is mounted to the articulated hub. A translation is made from the two rigid solids to the union point of the pistons. The piston is united to the solids by a joint with the z axis of rotation free, completing the assembly. The Bond Graph diagram of the blocks is represented in Figure 34.

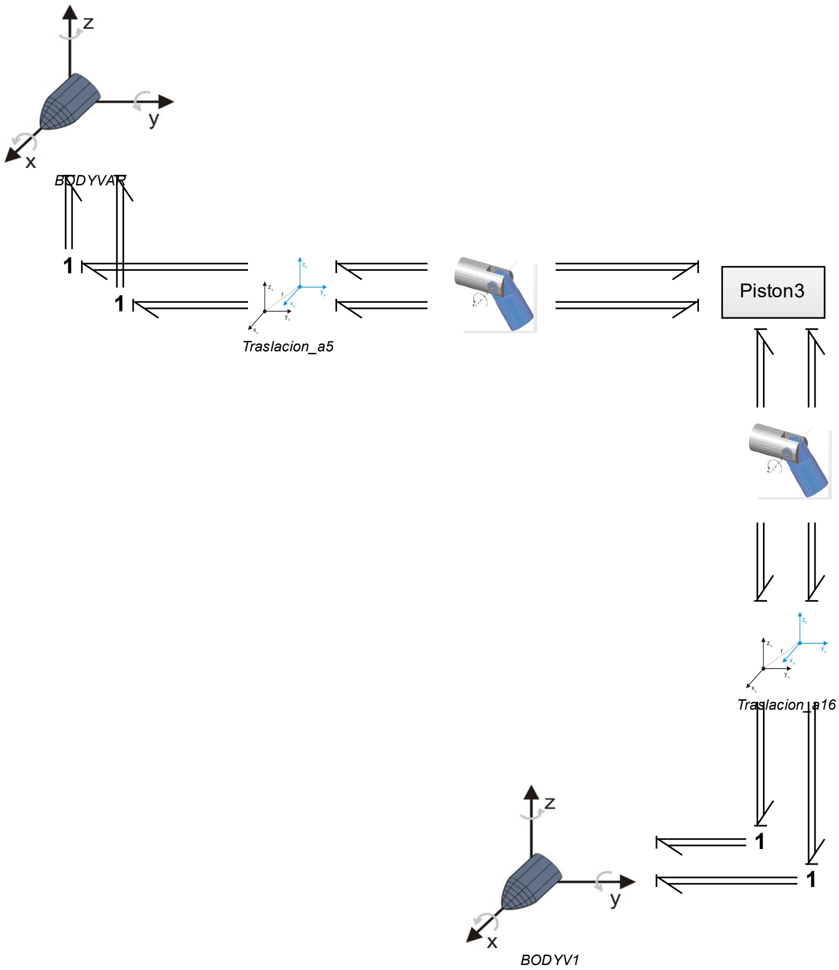


Figure 34. Front steering piston assembly diagram

As previously mentioned there exists a translation from the rigid solid to the joint. In the case of the main chassis the distance vector is the following:

```
D_V1_PP_K = PP_K - CM_V1;
```

In the case of the steering articulated hub, the distance vector is the following:

```
D_VAR_PP_L = PP_L - CM_VAR;
```

The main difference between this assembly and the central one is that the pistons in this case are in a different orientation from the main frame. To model this orientation, an initial angle has to be set in the rotation matrix inside the rotations in the joint. To do so, basically an initial value is given to the angle that will be updated. This is done in this case, but initial angles can be set in constrained angles of rotation too. The code inside the rotations is the following:

```
d_psi = psi_ini_P3_rad;
psi=int(dpsi,d_psi);
//teta=int(dteta);
//fi=int(dfi);
teta=0;
fi=0;
```

This is how the initial angle is set in the code, in the case of the rotation on the side of the pistons rod, the one that connects to the hub:

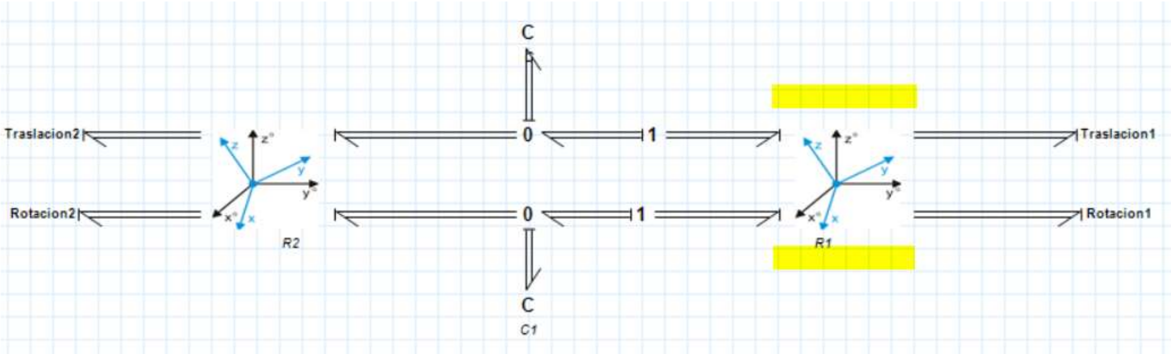


Figure 35. Modified rotation matrix in the rod side

And the angle value:

```
psi_ini_P3_rad = PsiP3_ini*(3.14/180);
```

Where:

```
PsiP3_ini = 97.54,
```

In the opposite side of the piston, the barrel part which is connected to the main frame, the angle is also set to the rotation connected to it:

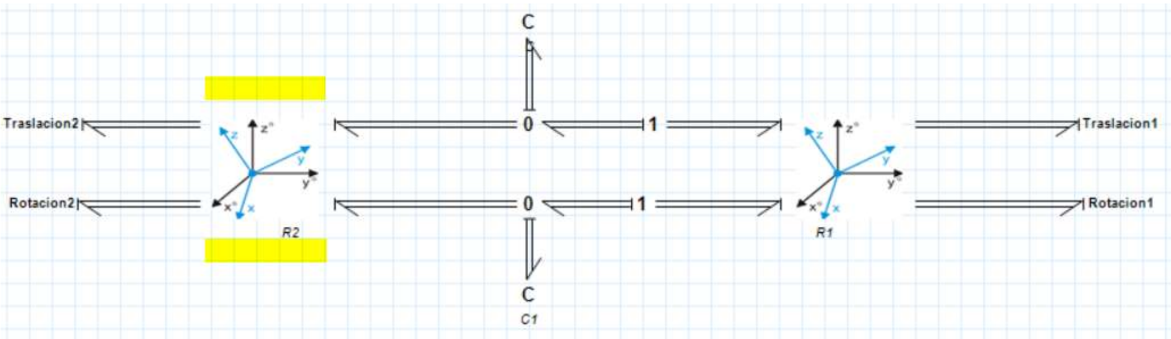


Figure 36. Modified rotation matrix in the barrel side

Where the modified code inside the rotation matrix is:

```
d_psi = psi_ini_P3_rad;
psi=int(dpsi,d_psi);
//teta=int(dteta);
//fi=int(dfi);
teta=0;
fi=0;
```

As it can be seen, both angles of rotation are the same. This orientates the axis of both rigid solids of the piston in the same direction.

And finally, the capacitors in the rotation side of the joint have set the z axis of rotation free:

```
parameters
 real global k;

variables
real a, b;
equations
a = constraint(p.f[1,1]);
b = constraint(p.f[2,1]);
//c = constraint(p.f[3,1]);
//    p.e[1,1] = 1e9 * int(p.f[1,1]);
//    p.e[2,1] = 1e9 * int(p.f[2,1]);
    p.e[1,1] = a;
    p.e[2,1] = b;
    p.e[3,1] = 0;
```

7.5.3 Front steering bar assembly

The next sub assembly of the front steering system is the union of the bar to the articulated hubs. This system is used to ensure that the wheels are not independent from each other, aiding in the control of the front steering. This Bond Graph diagram of this assembly is represented in Figure 37.

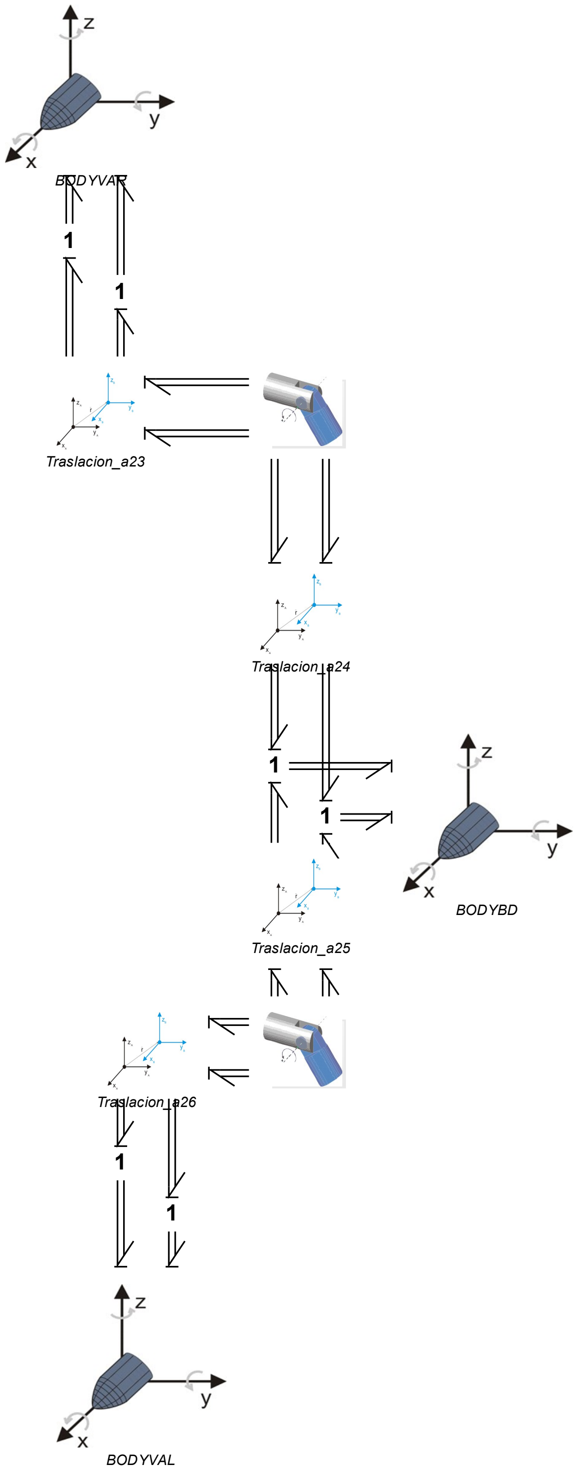


Figure 37. Steering bar assembly diagram

This diagram consists in the mounting of a rigid solid to two other ones. This assembly is done with joints leaving one axis of rotation free. From the centre of mass of the bar (BD), there is a translation to the union point between it and the articulated hub. In this union point there exists a joint that is set with different constraints in order to emulate the previously mentioned movement. Finally a translation is done from this joint to the centre of mass of the articulated hub, completing the assembly of the steering bar.

The translations from the centre of mass of the bar to the joint points are the following:

```
D_BD1_BD2 = BD_1 - BD_2;
D_BD3_BD2 = BD_3 - BD_2;
```

Where `BD_2` is the centre of mass of the bar, `BD_3` is the joint point in the right side and `BD_1` is the one in the left.

As previously mentioned, both joints leave free the z axis of rotation:

```
parameters
 real global k;

variables
real a, b;
equations
a = constraint(p.f[1,1]);
b = constraint(p.f[2,1]);
//c = constraint(p.f[3,1]);
//    p.e[1,1] = 1e9 * int(p.f[1,1]);
//    p.e[2,1] = 1e9 * int(p.f[2,1]);
    p.e[1,1] = a;
    p.e[2,1] = b;
    p.e[3,1] = 0;
```

So the `psi` angle has to be updated:

```
psi=int(dpsi);
//teta=int(dteta);
//fi=int(dfi);
teta=0;
fi=0;
```

Finally, there is a translation from the joint to the centre of mass of the articulated hub, where the distance vectors are the following:

```
D_FR_BD3 = BD_3 - CM_VAR;
D_FL_BD1 = BD_1 - CM_VAL;
```

When constraining one movement from two or more joints the use of the constraint function can lead to instability in the simulation. In some cases the simulations can halt and give the user and error. In this case the recommendation is to give a high value of a constant inside the capacitor. This solution will increase the total simulation time, but it will allow its continuation or realization. This method of setting a high value constant can be seen in the commented code in green, in the majority of examples showing the code inside the capacitor elements in the different joints of the model.

7.5.4 Complete front steering assembly

Finally the complete assembly of the front steering system can be observed in Figure 38.

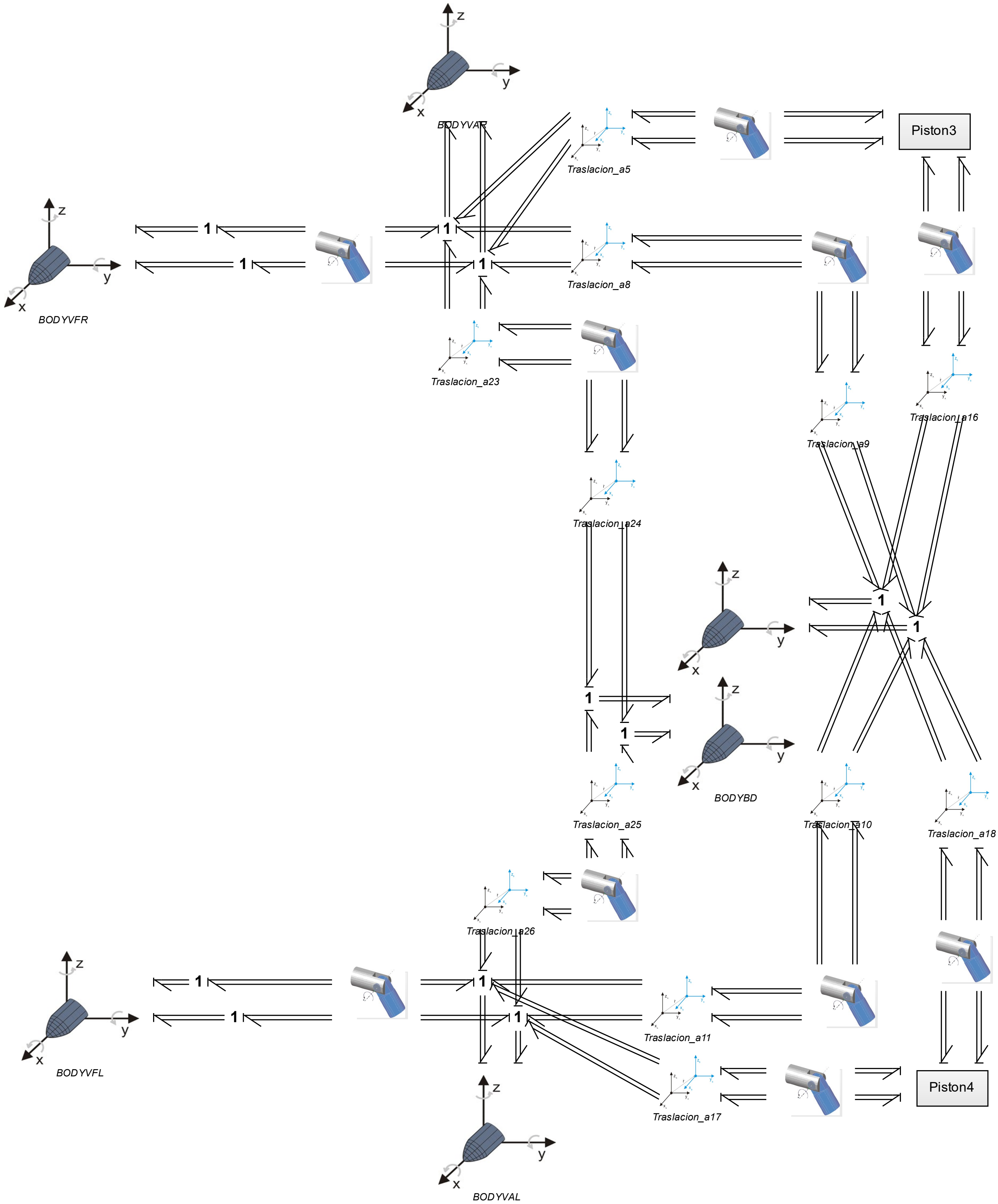


Figure 38. Front steering assembly

7.6 Tyre contact model

A vital point for obtaining a correct simulation is to model correctly the contact between the tire and the ground. This model has to include the dampening and elastic characteristics of the tire, enabling the wheel to give realistic responses to bumps and uneven terrain. The other main point in the modelling of the tire is the reaction forces of the ground, especially the one in the traction direction. There exists different ways to model the tire with Bond Graph [16] [17]. The block assembly of the tire model created is represented in Figure 39 [15].

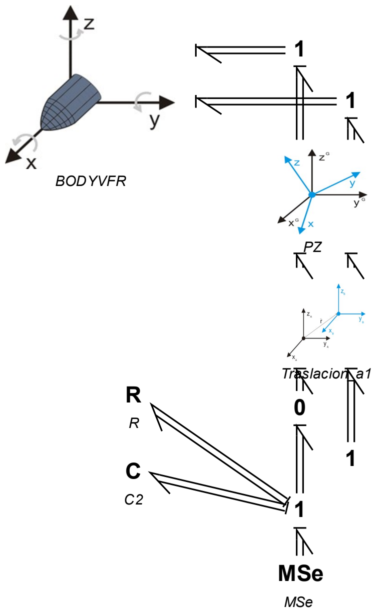


Figure 39. Tyre contact assembly diagram

This Bond Graph sub-assembly consists in a rotation followed by a translation, with its distance vector being the distance from the wheels centre of mass and the ground. The rotation is used to ensure that the translation vector will always point to the ground. If it was not there, the distance vector would rotate with the wheel. Ignoring the `0` junction, the ground is defined by two `1` junctions where all the reactions that model the ground will be applied. Since they are forces, they are applied to the translation power transfer.

For the vertical reactions, in the z axis, a dampening and an elastic constant are set to enable the tractor to make contact to the ground and not sink. For the lateral reaction another dampening is set in order to minimize the slip in this direction. This is done because the modelled vehicle is a tractor, and in the speed in which it will operate the lateral reaction does not have to be precisely calculated. The main job of the lateral dampening is to stop the vehicle from sliding. If the user of the model wanted to simulate the vehicle in frozen or sowed conditions, this dampening would have to be eliminated, and the reaction should be calculated like the one in the x direction. The capacitor and the resistors values are the following:

For the resistor:

```
parameters
    real r[3,3] = [0.0, 0.0, 0.0; 0.0, 1000.0, 0.0; 0.0, 0.0, 10000.0];
equations
    p.e = r * p.f;
```

For the Capacitor:

```
parameters
   real global k;
|
equations
//p.e = constraint (p.f);
// p.e = 1e9 * int(p.f);
//   p.e[1,1] = 1e9 * int(p.f[1,1]);
//   p.e[2,1] = 1e9 * int(p.f[2,1]);
//   p.e[3,1] = 1e9 * int(p.f[3,1]);
   p.e[1,1] = 0;
   p.e[2,1] = 0;
   p.e[3,1] = 1e8 * int(p.f[3,1]);
```

As previously mentioned, the reaction in the traction direction of the wheel, x, has to be calculated in order to give an accurate model of acceleration and breaking. To calculate this reaction the Pacejka model is used, aided by the 0 junction from where the vertical reaction of the ground, in z, will be obtained.

7.6.1 Pacejka model

The Pacejka model enables the calculation of the forces and moments resulting from the tire contact to the ground [15] [18] [19]. For any simulation is almost vital to obtain realistic results of the behaviour of a vehicle. Even though in this model only the longitudinal force is calculated, it is important to explain the lateral force and the aligning moment since they will be easily added to the modelling block.

The Pacejka model obtains the forces of the tire thanks to a set of equations and coefficients. Each of the two forces and the moment will have a set of factors to obtain their values:

- B is the stiffness factor
- C is the shape factor
- D is the peak value
- E is the curvature factor
- S_h is the horizontal shift
- S_v is the lateral shift

These factors will be obtained through a set of constants that will depend on what force or moment is calculated. These constants are named as a_i.

Lateral force (Fy)

$$C = 1{,}3\ ;$$

$$D = a_1 {F_z}^2 + a_2 F_z\ ; \tag{36}$$

$$BCD = a_3 \sin(a_4 \tan^{-1}(a_5 F_z))\ ; \tag{37}$$

$$B = \frac{BCD}{CD}\ ; \tag{38}$$

$$E = a_6 {F_z}^2 + a_7 F_z + a_8\ ; \tag{39}$$

The shift factors are slightly affected by a camber angle γ

$$S_h = a_9 \gamma\ ; \tag{40}$$

$$S_v = \left(a_{10} {F_z}^2 + a_{11} F_z \right)\gamma\ ; \tag{41}$$

And finally, the lateral force is obtained in function of the slip angle α:

$$\phi = (1 - E)(\alpha + S_h) + \frac{E}{B}\tan^{-1}\big(B(\alpha + S_h)\big) ; \tag{42}$$

$$F_y = D\sin(C\tan^{-1}(B\phi)) + S_v \ ; \tag{43}$$

Aligning moment (M_z)

$$C = 2{,}4 ;$$

$$D = a_1 {F_z}^2 + a_2 F_z ; \tag{44}$$

$$BCD = \frac{a_3 {F_z}^2 + a_4 F_z}{e^{a_5 F_z}} ; \tag{45}$$

$$B = \frac{BCD}{CD} ; \tag{46}$$

$$E = a_6 {F_z}^2 + a_7 F_z + a_8 ; \tag{47}$$

The shift factors are slightly affected by a camber angle γ

$$S_h = a_9 \gamma ; \tag{48}$$

$$S_v = \left(a_{10} {F_z}^2 + a_{11} F_z \right)\gamma ; \tag{49}$$

And finally, the lateral force is obtained in function of the slip angle α:

$$\phi = (1 - E)(\alpha + S_h) + \frac{E}{B}\tan^{-1}\big(B(\alpha + S_h)\big) ; \tag{50}$$

$$M_z = D\sin(C\tan^{-1}(B\phi)) + S_v \ ; \tag{51}$$

Longitudinal Force (F_x)

$$C = 2{,}4 ;$$

$$D = a_1 {F_z}^2 + a_2 F_z ; \tag{52}$$

$$BCD = \frac{a_3 {F_z}^2 + a_4 F_z}{e^{a_5 F_z}} ; \tag{53}$$

$$B = \frac{BCD}{CD} ; \tag{54}$$

$$E = a_6 {F_z}^2 + a_7 F_z + a_8 ; \tag{55}$$

And finally, the lateral force is obtained in function of the slip s:

$$\phi = (1 - E)s + \frac{E}{B}\tan^{-1}(Bs) ; \tag{56}$$

$$F_x = D\sin(C\tan^{-1}(B\phi)) \ ; \tag{57}$$

The values of the constants can be found in the following table:

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
F_y	-22,1	1011	1078	1,82	0,208	0	-0,354	0,707	0,028	0	14,8
M_z	-2,72	-2,28	-1,86	-2,73	0,11	-0,07	0,643	-4,04	0,015	-0,07	0,945
F_x	-21,3	1144	49,6	226	0,069	-0,006	0,056	0,486	0	0	0

Table 5. Constants of the Pacejka model

As previously mentioned, since the vehicle is going to move at slow speeds only the longitudinal force is introduced in the model. To do so, two blocks are created, one to calculate the slip of the tire and the other to calculate the force.

The Slip Block is the following:

```
parameters
   real r_tire=0.423;
variables
   real v_tire, v_car;
equations
   v_tire = w_tire[2,1] * r_tire;
   v_car = v_chassis[1,1];
   if v_tire>v_car and v_tire<>0 and time>0.2 then
      slip = (v_tire-v_car)/v_tire * 100;
   else
         slip = 0;
   end;
```

The velocity of the vehicle is obtained with a signal coming from the `V1` translation `1` union, and the angular velocity of the tire is obtained with a signal from the rotation `1` union of the wheel. With the two signal values, the slip is calculated [15].

$$slip = \frac{v_{tire} - v_{car}}{v_{tire}} * 100\,; \qquad (58)$$

Once it is calculated, its value is sent to the Pacejka block to calculate the longitudinal force. To this block, a signal of the vertical force in the contact point (Fz) is sent:

```
parameters
   real a1=-21.3, a2=1144,
    a3=49.6, a4=226, a5=0.069,
    a6=-0.006, a7=0.056, a8=0.486;
variables
   real F, D, C, B, E, X, PHI, FZ, BCD;
code
   // start typing here
   X = input;
   FZ = port1[3,1]/1000;
   D = a1 * FZ^2 + a2 * FZ;
   BCD = (a3*FZ^2+a4*FZ)/exp(a5*FZ);
   C = 1.65;
   if C<>0 and D<>0 then
   B = BCD / (C*D);
   else B=0; end;
   E = a6*FZ^2+a7*FZ+a8;
   if B<>0 then
   PHI = (1-E)*X+(E/B)*arctan(B*X);
   else PHI = 0; end;
   F = D * sin(C*arctan(B*PHI));
   output = F;
```

The outlet force is sent to the modulated source effort on the tire contact point, represented in Figure 40.

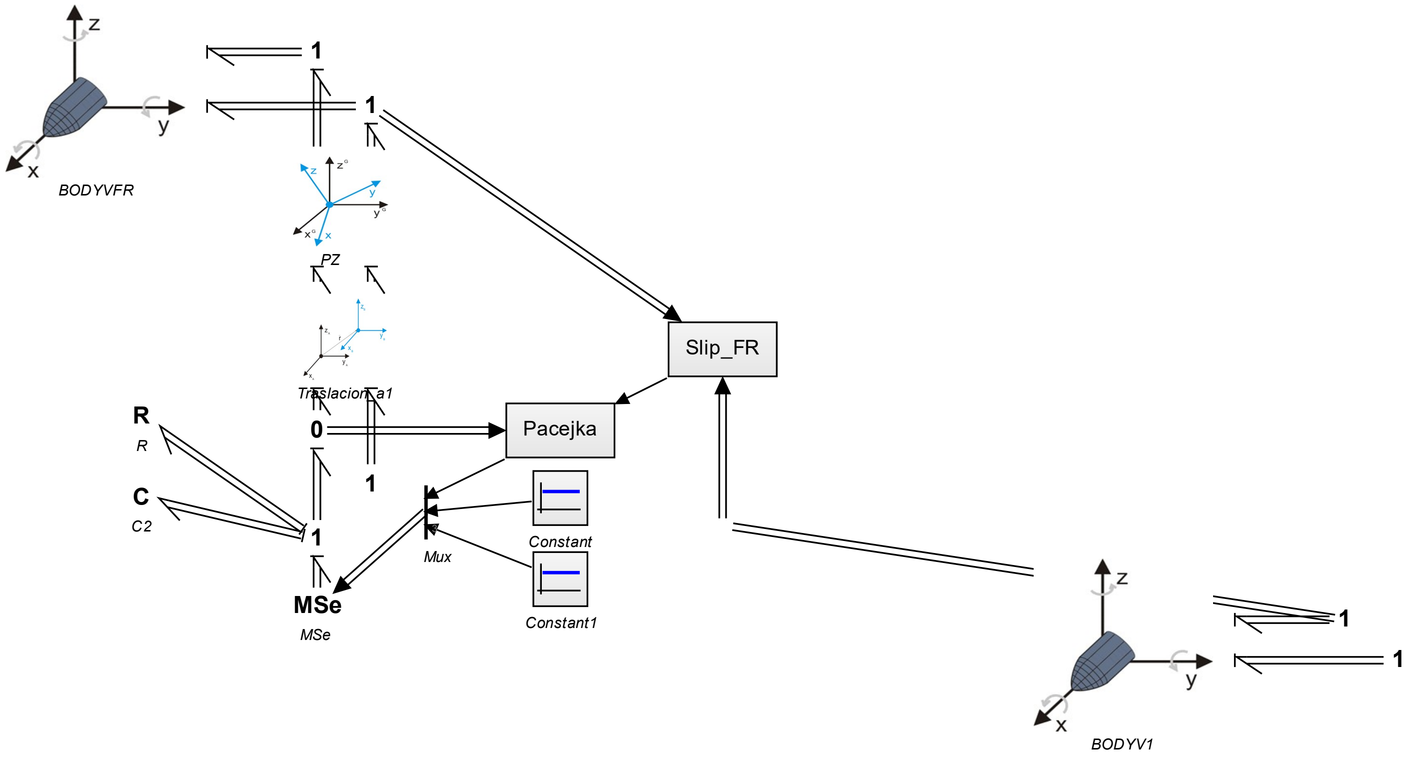
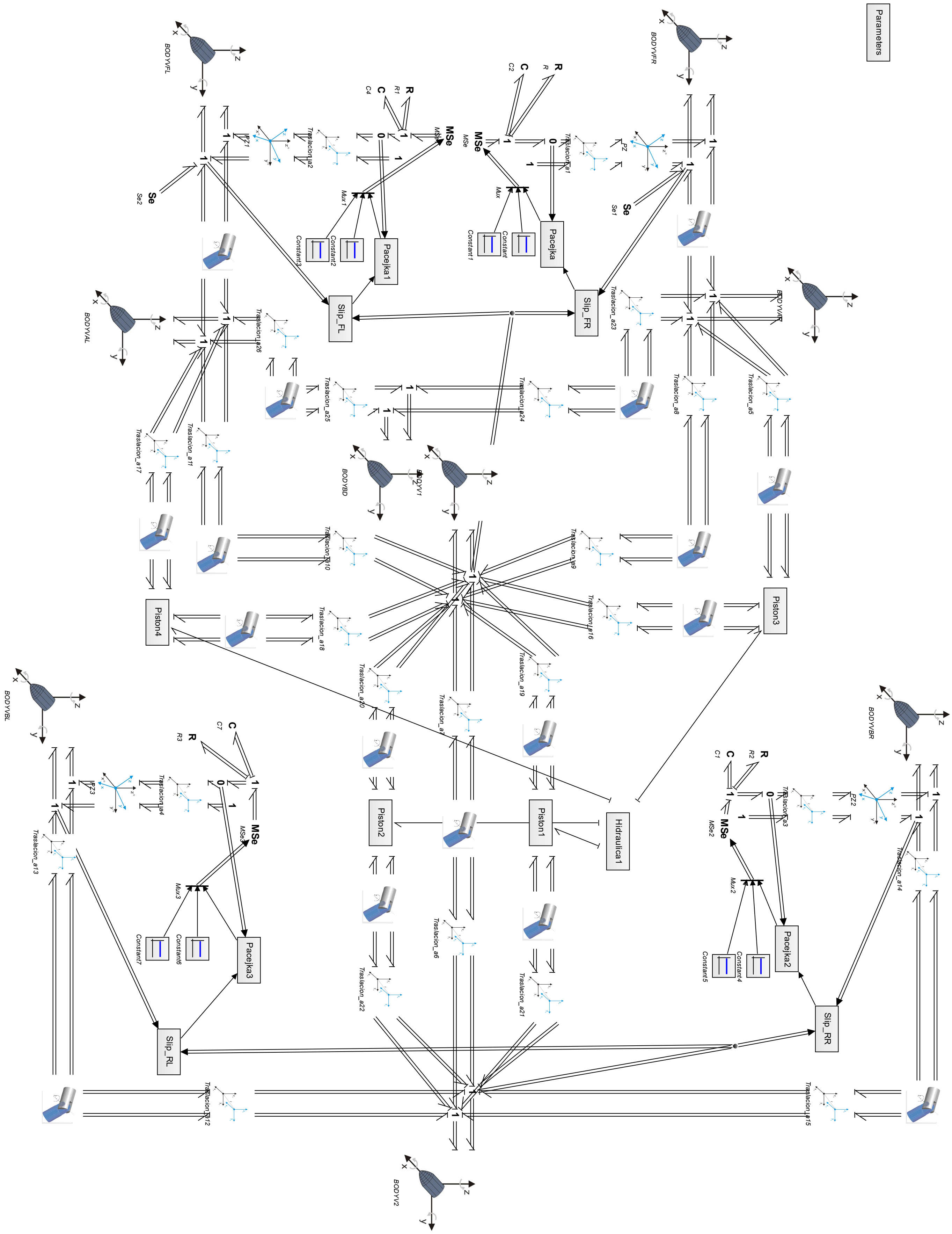


Figure 40. Pacejka model block diagram

The other constants that enter the mux are set to zero. Instead of the y constant the lateral force could be added, and in the rotation `1` union the aligning moment could also be added with a mux and a modulated effort source, the same as done for the longitudinal force. The strategy would be to calculate the slip angle with another block, send it to the Pacejka one and calculate the two forces and the moment inside it. Once calculated, the block would send the two force signals to one mux + modulated source and the moment to another mux + modulated source.

7.7 Complete Bond Graph model



7.8 Animation and Result Plots

Once the model has been assembled, the next step before starting a simulation is to create a set of plots in order to track and record the behaviour of the created model. 20Sim enables its users the creation of such systems, mainly divided in two groups. The first one is the typical plots, where normally a value of interest is tracked during the simulation, giving a plot of its numerical values for every recorded time step. This is helpful to record the positions, pressures, reaction forces, etc. The main values of interest are the positions of the bigger rigid bodies of the model, the two main volumes and the wheels. Knowing their position aids in obtaining their trajectory, the orientation of the vehicle, etc.

The second group of simulation results is the animation plots. This type of plot basically binds a 3D object to a reference frame, and this reference frame is dependent on a set of input values from the simulation. In the case of this project, the reference frame is dependent on the recorded movements and rotations of a rigid body of the model. The 3D objects can either be imported from a CAD software or created with 20Sim. The creation is limited, so complex objects should be made with anther software.

It is important to know that the values that appear in the plots can be saved in a .csv file, giving the user the ability to extract further results from the recorded data. So if the user wants to save the values of a variable during the time span of the simulation, it is important that this variable is recorded either in a 2D plot or in a 3D animation.

7.8.1 2D plot

The main 2D plot used throughout the creation of the model has been the plotting of the 3 values of the position vector of the main solids of the model. To do so a plot is created, and the three values of the q sensor are set in the y axis:

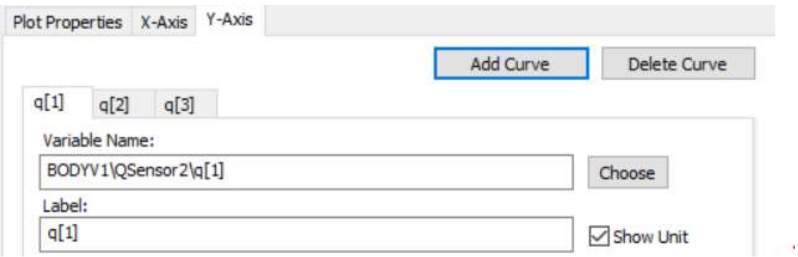


Figure 41. Plot properties.

As seen in Figure 41 any number of curves can be added to a plot. Here a plot is created for each solid with three curves representing the position vector. This plot is really useful during the creation of the model to check if the behaviour of each solid is normal or correct.

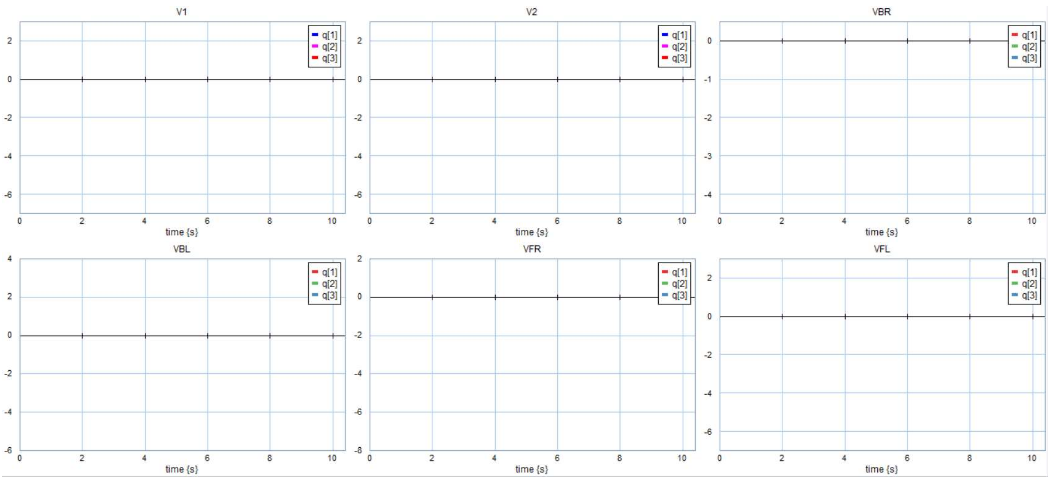


Figure 42. Rigid solid positions plot

In this plot the x axis is always set to be the time, but it does not have to be. This setting in the x axis is recommended to record the values in a csv file, since all of them will appear ordered with the time of each recording. Once the model works, a nice plot to observe is the 2D trajectory of the tractor. To do so, a plot has to be created with a curve representing the y position and the x axis representing the x position of one of the solids. The chosen one is the front one, `V1`. To do so, the variable time is changed for the variable of the x value recorded by the `qsensor` of the rigid solid 1. This enables the user to have the recorded trajectory of the vehicle.

Figure 43. Trajectory plot setting 1

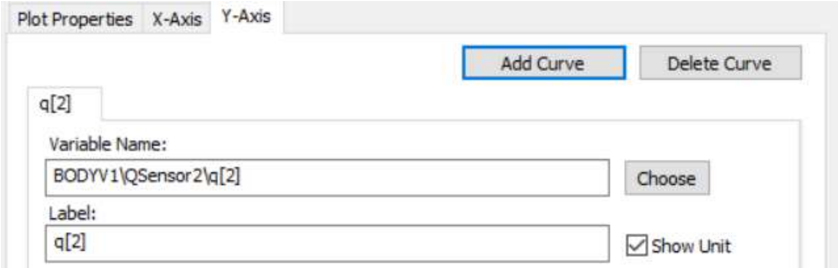


Figure 44. Trajectory plot setting 2

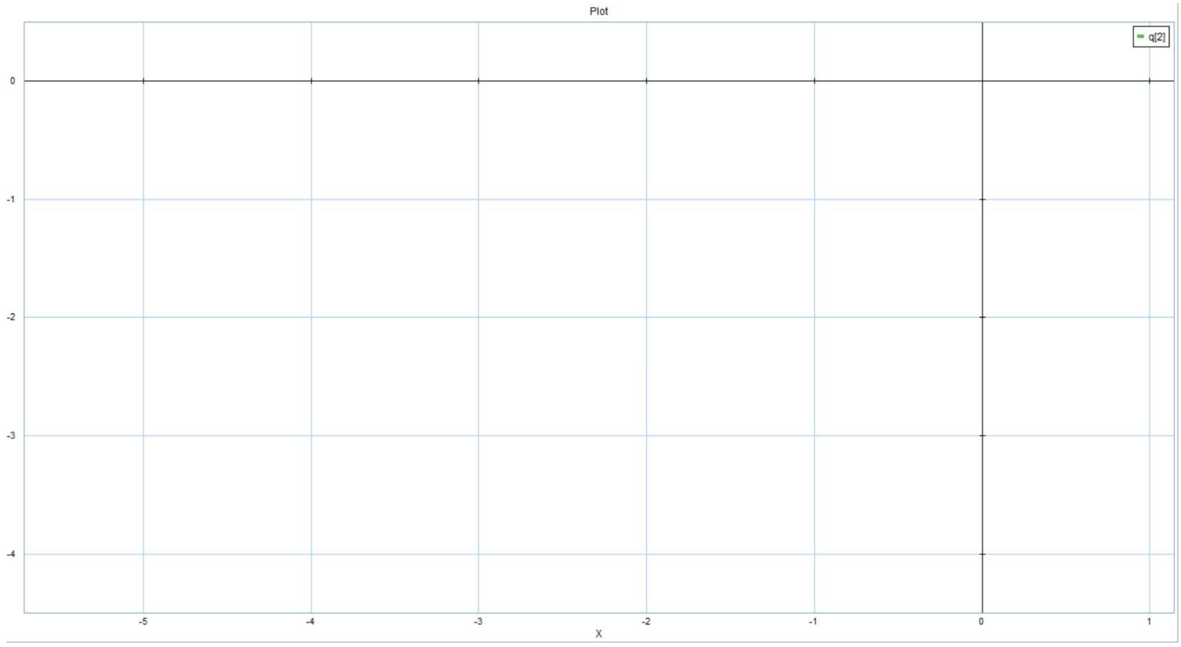


Figure 45. Space where the trajectory plot will appear

The other important plot to include is the one that records relevant values of the hydraulic system. This can include the position of the pistons, the pressure applied to them, the pressure of the pump, etc. In this case, the position of the pistons and the pressure on the pump is recorded.

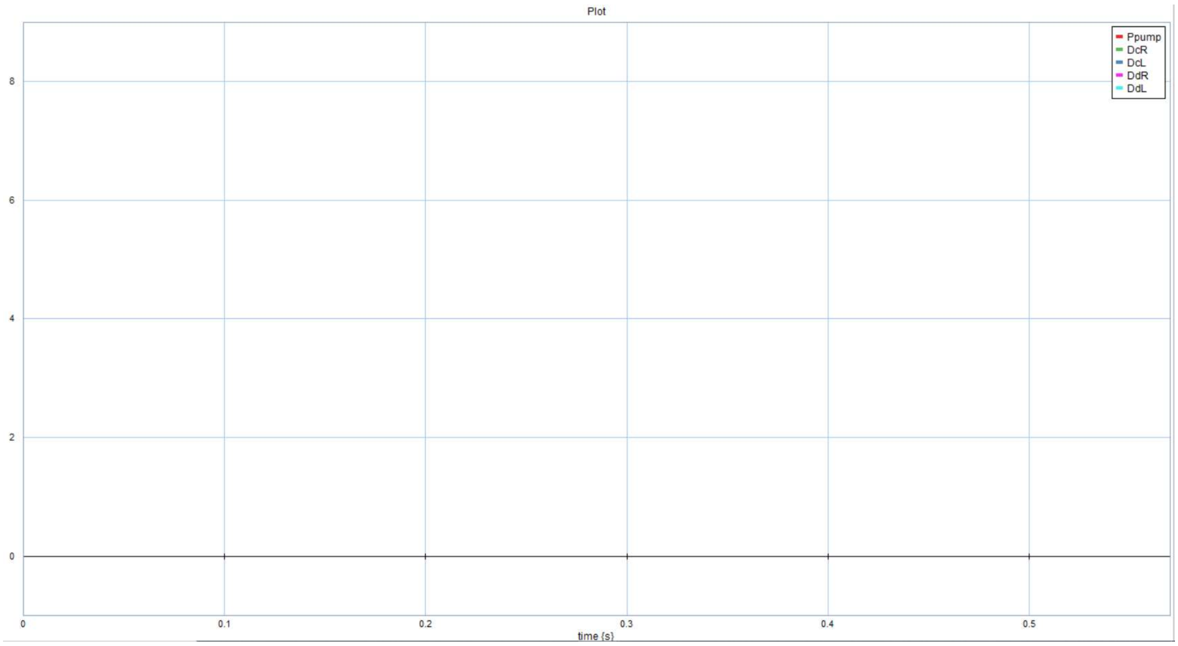


Figure 46. Space where the hydraulic system plots will appear.

7.8.2 3D animation

The final step for recording the simulation is to create a 3D animated model using an animated plot window. To do so, the user has to enter into the properties of the animation, and create a reference frame for each of the solids to be represented in the animation. All of the solids do not have to be represented, just the more interesting ones. In each reference frame, the variable of the `qsensor` of each body is set in each of the positions:

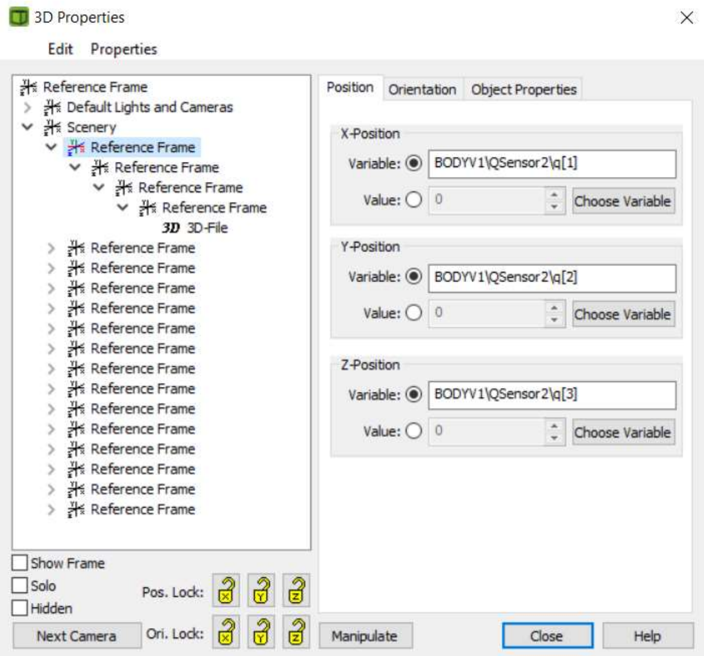


Figure 47. 3D animation position setting

Once the positions are set, three new reference frames have to be created, one inside the other like in Figure 47. In each of these reference frames one of the three angles is going to be updated. This gives better precision than if they were all updated in the initial reference frame where the positions are set:

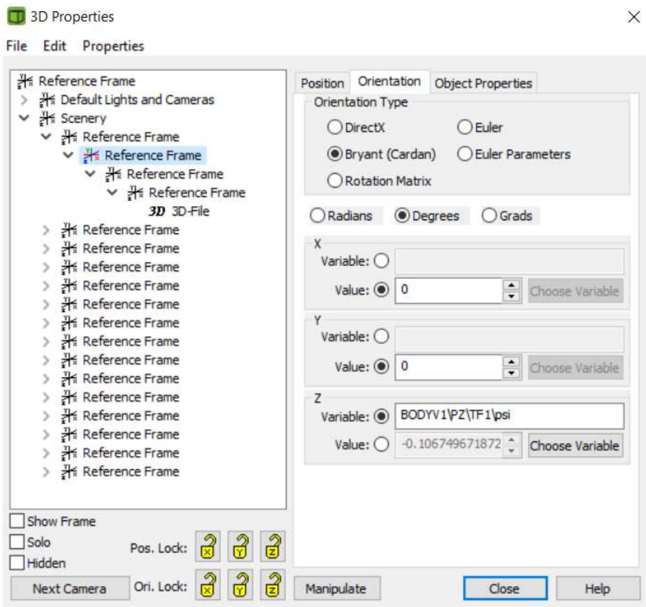


Figure 48. 3D animation angle setting

Once all the angles are set, in the last reference frame the CAD or the 20Sim created object is introduced, appearing in the 3D space of the animation. This is done for all the relevant rigid solids. The only ones not included are the articulated hubs. The resulting animation is minimalistic, but this aids in the performance of the simulation.

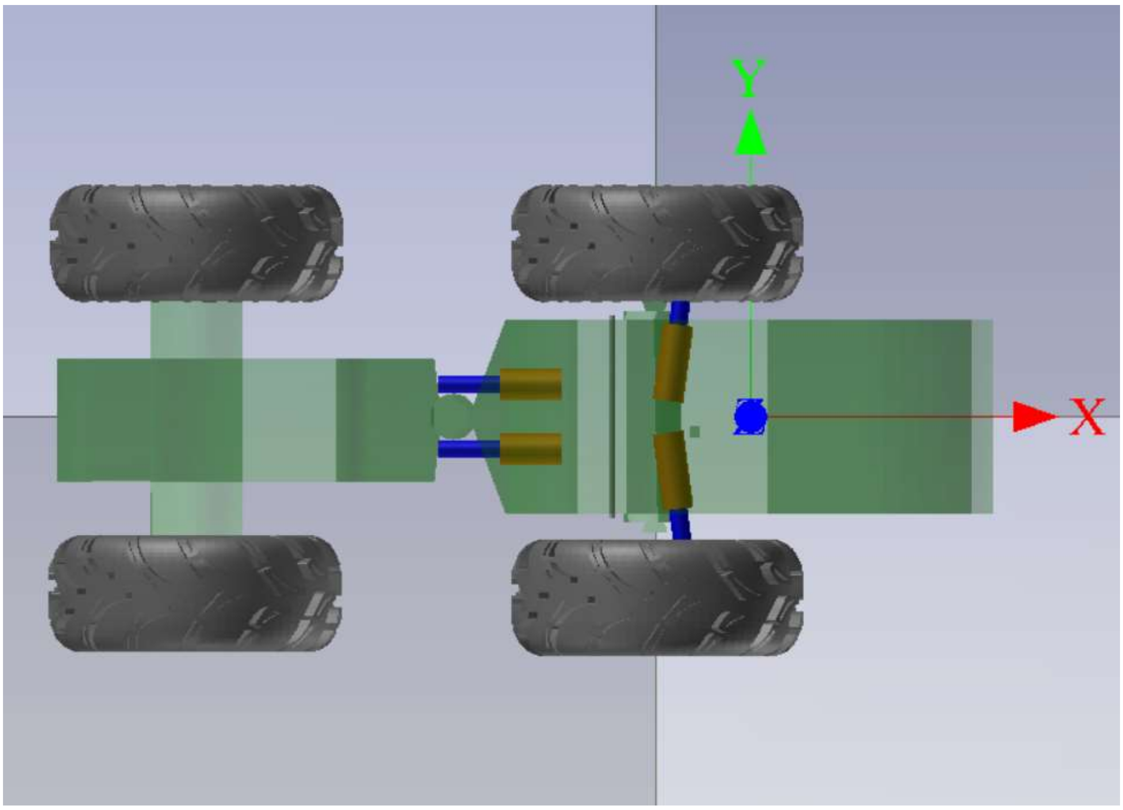


Figure 49. 3D animation model - Top view

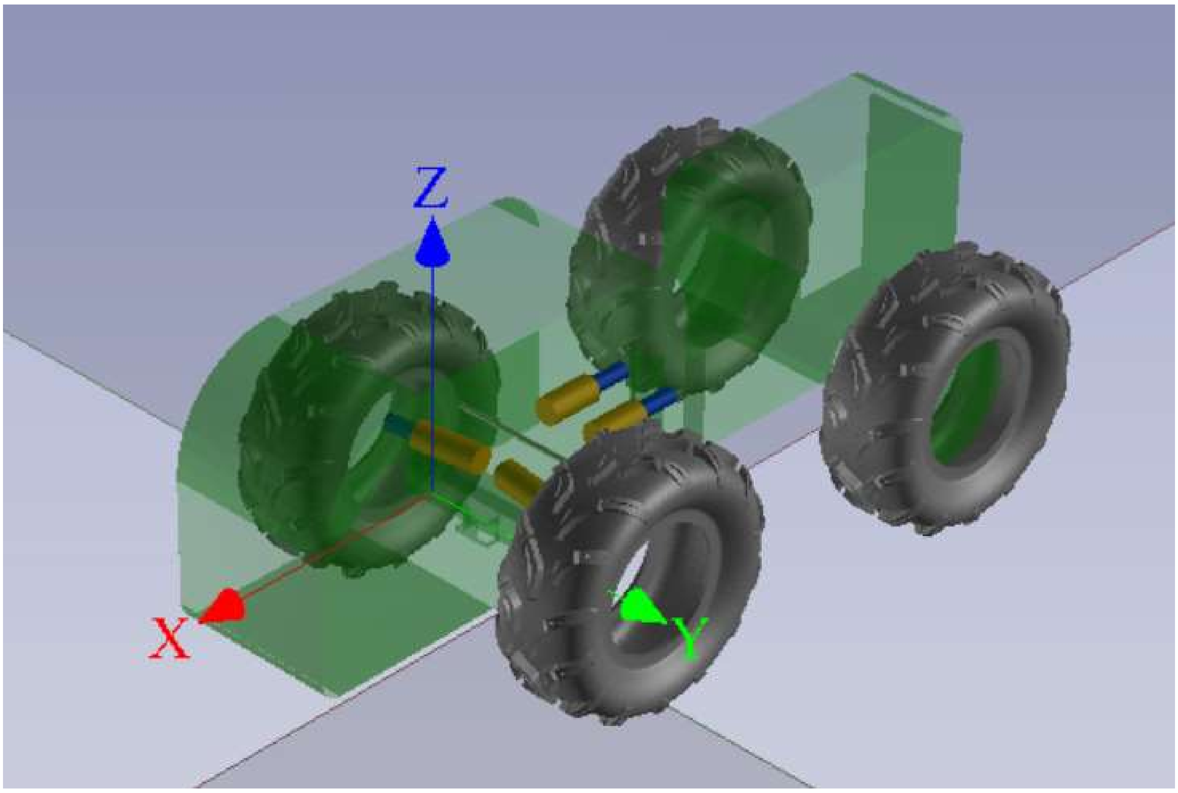


Figure 50. 3D animation model - isometric view

The main objective of the animation is to have a visual representation of the tractor and its moving parts, mainly the pistons and the wheels. Because of this the simple shape given to the main volumes has been set to a transparency that allows the visualization of the front and central steering systems. Without the transparency the pistons and the steering bar would not be visible.

In this case, both the main volumes have been done with solid works. They are a simplified visualization of an articulated tractor, almost like a toy one, but these 3D CADs are simple enough to represent the main dimensions of the vehicle without adding a significant load to the computer while doing the animation.

7.9 Hydraulic Systems

The hydraulic systems used to test the vehicle model have been provided by the tutors of the project. Two systems have been tested, a simple one composed of two switch valves, one for the front and another for the central steering, and another more complex one.

The simple one consists of a switch valve that connects to two pistons, directing flow to one side or the other. With this system there will be a block for each steering system, one for the front and another for the back. Each block will be composed of the diagram in Figure 51.

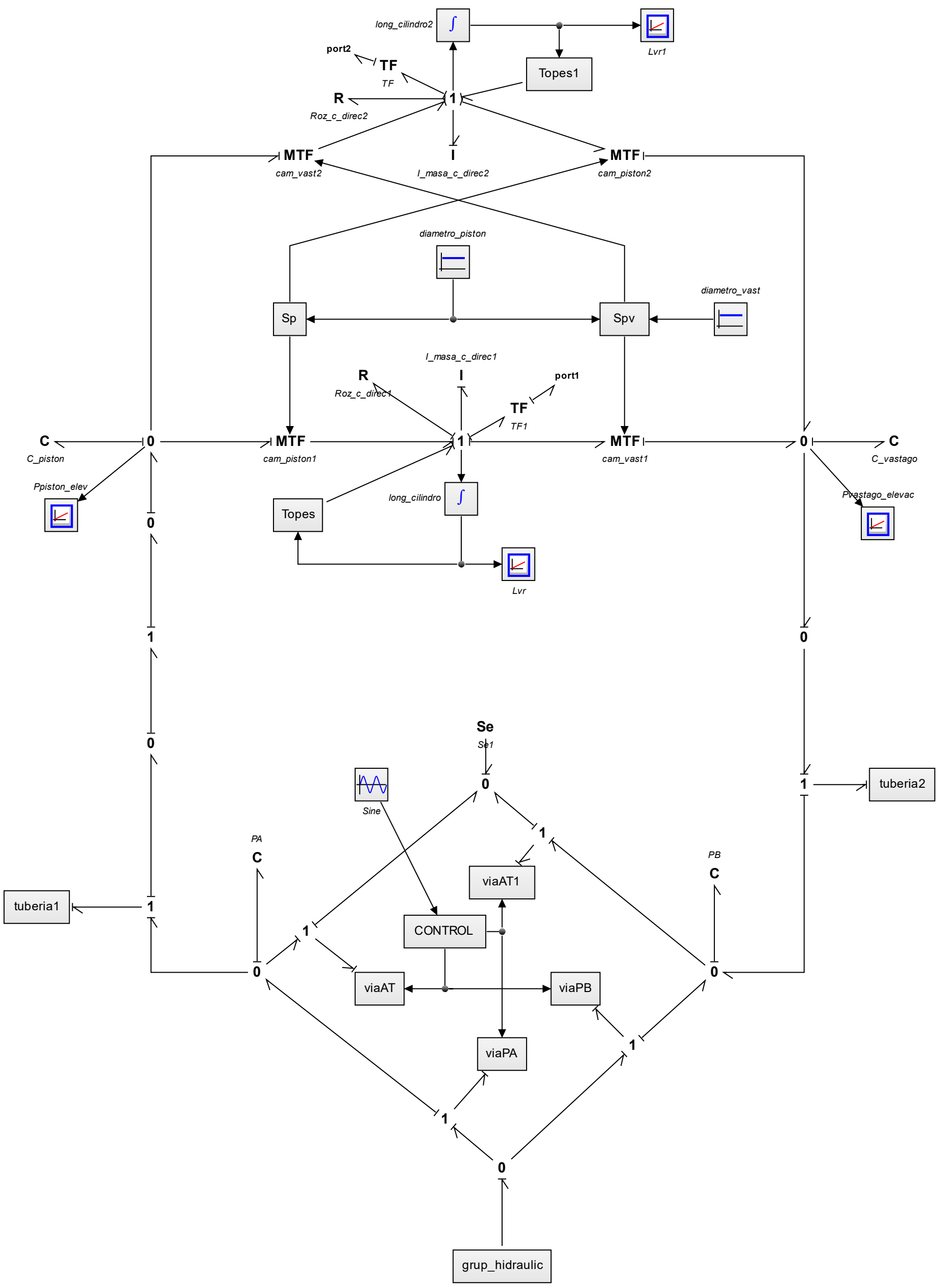


Figure 51. Simple hydraulic system diagram

The simplified hydraulic diagram is represented in Figure 52.

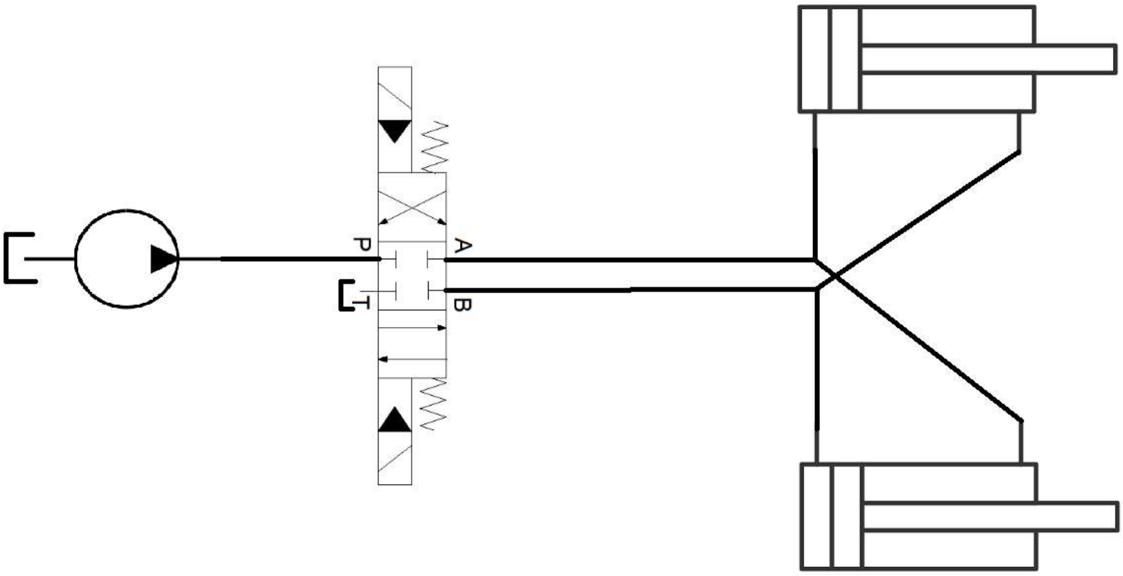


Figure 52. Simplified hydraulic diagram

This system basically directs the flow and pressure created in the `grup_hidraulic` block to one side or the other of the graph. The control blocks in the middle connect one of the two vertical channels to the pump block and the other to the tank, which is represented with an effort source with negative pressure [10].

To switch between the two connections, a signal is sent. When it is higher than 0, flow and pressure go to the left. When is smaller than 0, the opposite happens. In both the two vertical channels going to the piston exists a block called `tuberia`, which defines the area of the tube, setting the flow of oil for a given pressure. This flow is sent to two pistons, which are the diagrams on the top.

These diagrams basically multiply the flow and pressure by the surface of the pistons, different for each side due to the rod, obtaining the force and the velocity with the modulated transformers. The transformers send the force and velocity to a `1` junction, where the inertia and friction of the piston are set. There is a signal coming of the `1` junction where the velocity is integrated, adding it up to the initial elongation and giving the position of the piston. This position is sent to the block `Topes`, where it will be tracked, and if it surpasses the maximum and minimum elongations, an effort will be sent to the one junction emulating a stopping force. The signal used is a sinusoidal one, which will make the tractor steer from one side to the other.

From the `1` junction, an output power transfer will go to a transformer that will multiply by 0.01, transforming from HeN cm/s to N m/s, since the first ones are the units of the hydraulic Bond Graph. In the case of the central articulation, since both pistons are oriented in the same direction, the left transformer has to be negative, -0.01, in order to correctly send the velocity flow to the multi solid model. Each of the four power transfer elements will go to the prismatic joint of its piston, joining together the hydraulic system and the multi solid model.

The multi solid model will be tested with this hydraulic system since the more complex one didn't work during the realisation of the project. Even though its implementation was not possible, it is important to do a brief explanation on its diagram since it is a really close representation of a real hydraulic system for the articulated tractor.

The complex hydraulic system is made to emulate a real hydraulic system for the tractor. The overall system is composed by a set of five main elements, which are: The hydraulic side of the piston, the pump, a flow diverter valve, an `orbitrol` valve for the front steering and a switch valve for the central one. The Bond Graph diagram is represented in Figure 53.

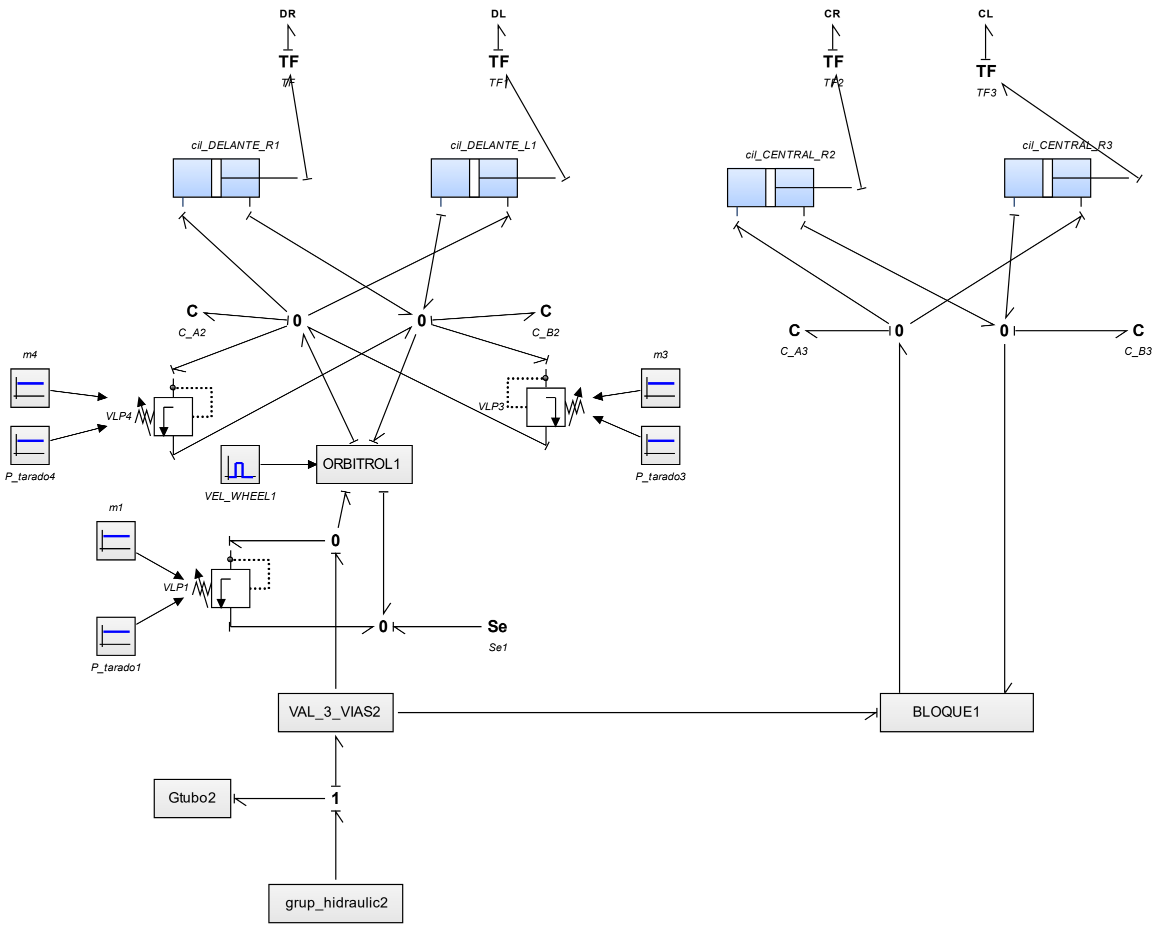


Figure 53. Complex hydraulic system diagram

The `grup_hiraulic2` represents the pump, and is the block that sends the oil flow to the system. This is done by multiplying the rpms of the engine to the displacement of the pump. The pumped flow goes to a `1` junction where a block is connected called `Gtubo2`. This block sets the flow from the pump to the valve by calculating from the pressure at the `1` junction and a defined area inside the block. The next block, `VAL_3_VIAS2` is in charge of dividing the flow coming from the pump to each of the two steering systems.

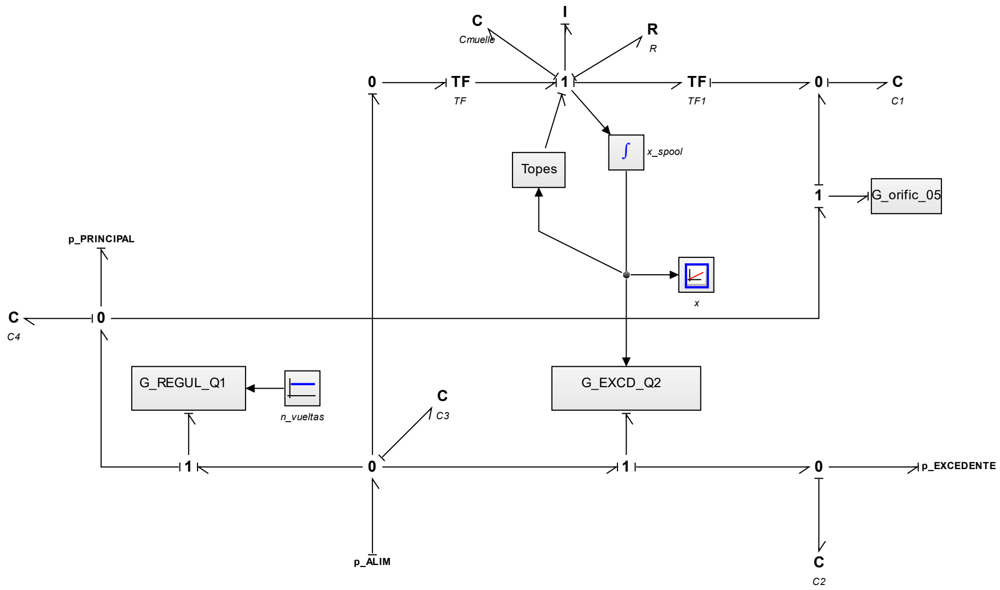


Figure 54. Flow diverter valve

This valve basically has a piston represented by the top `1` junction. This piston moves depending on the pressure and flow that enters the valve, and by moving it modifies the open surface area of the tube that goes to the front articulation, called `p_EXCEDENTE`, on the bottom right. The flow that does not go through this open surface goes to the central steering system through `p_PRINCIPAL`. This way the flow is divided, making more of it go to the central pistons since they will endure a greater load.

The `orbitrol` valve consists in a set of small holes that open when the wheel turns, letting flow through. When the wheel stops turning, these holes are shut. This is a typical valve mounted in heavy machinery to fulfil this function. Its Bond Graph is represented in Figure 55.

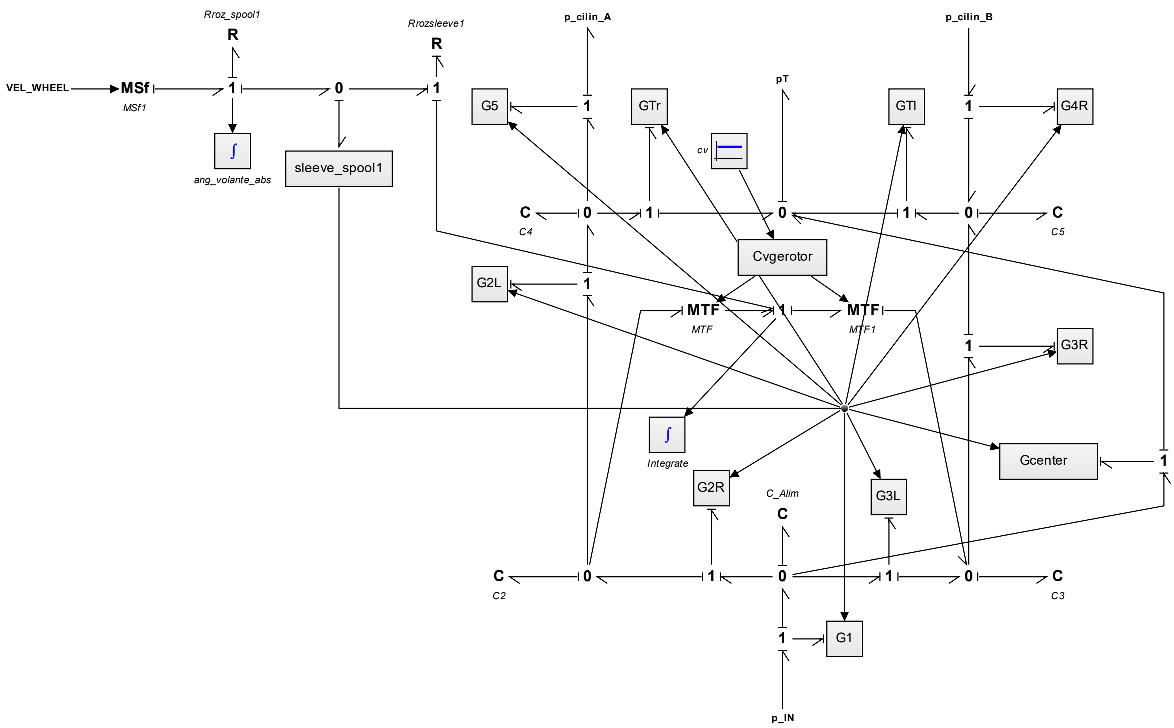


Figure 55. Orbitrol Bond Graph

The flow enters from the bottom of the graph, through `p_IN`. The angle of the wheel enters from the top left, and is the wheel input that the user gives. Two main paths can be seen from the input of the flow, one that goes to the left and up to `p_cilin_A`, which is the output for one side, and another path to the right and up to `p_cilin_B`, which is the output of the other side. Throughout each path there are `1` junctions connected to blocks with the letter `G`, and `0` junctions connected to a Capacitor. The `0` junctions and the capacitor represent losses due to the compressibility of the oil. The `1` junctions and the `G` block set the flow through each path in a similar manner as the tube explained previously. The main difference is that for this case exist a set of tables, one for each hole, that determine the open area of the hole for a given angular speed of the wheel. It can be seen how a signal coming from the wheel velocity is sent to the middle of the graph and then sent to all the holes.

Once the velocity of the wheel arrives to the hole blocks, 20Sim access the excel tables where an area is set for that angular speed, and obtains the oil flow with a function dependent on the area of the hole. Between each hole block there is a `0` junction and a capacitor to track the compressibility of the fluid. For the central block the Bond Graph diagram is represented in Figure 56.

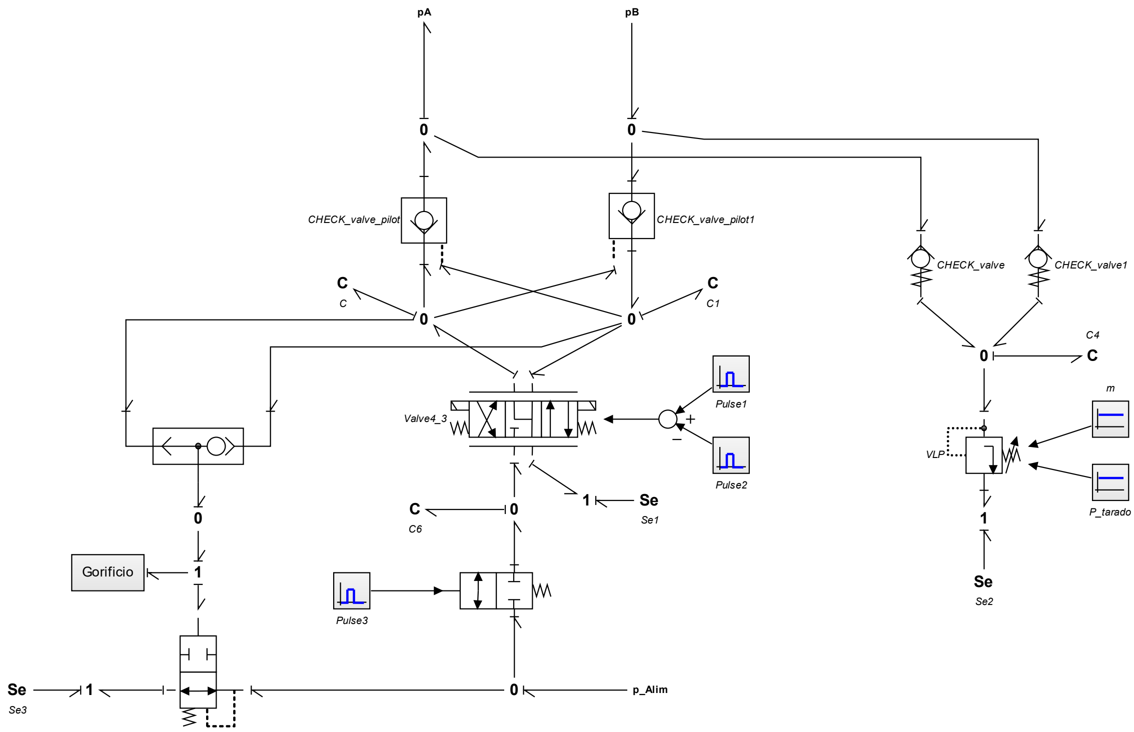


Figure 56. Central switch valve diagram

Without entering in the safety release and anti-return valves, the importance of this block is the switch valve in the middle, where pulses are sent. This valve sends the oil to a selected conduct to each piston, `pA` or `pB`. Depending on the input value, 1 or -1, the valve directs the input flow to one channel, and the flow coming from the other is sent to the hydraulic tank. This is done by setting a negative pressure to a `1` junction with a source flow `Se`. The rest of the valves add complexity and represent the safety system of a real hydraulic system, but the main function of the whole block is to direct flow to one channel or the other with a pulse of 1 or -1.

The final piece of the hydraulic system is the hydraulic side of the piston represented in Figure 58.

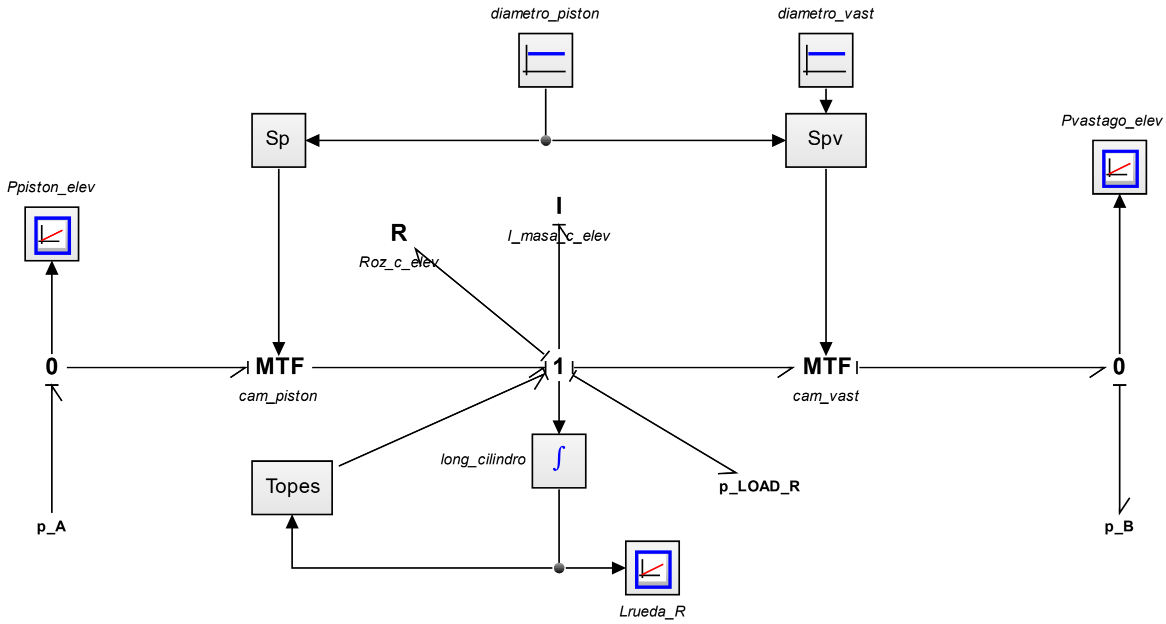


Figure 57. Piston hydraulic Bond Graph diagram

Depending on where the orbitrol or switch vale have directed the flow, it enters on one side and its transformed with a modulated transformer from pressure and flow to force and velocity by multiplying by the surface of the piston. Depending on the side, this surface varies since in one side there is the rod, so there is a smaller surface. The velocity coming from the `1` junction is tracked and integrated to know the distance moved by the piston. This distance is sent to the `Topes` block, where when it arrives at the maximum or minimum elongation a high force will be sent to the 1 junction to stop and block the piston. A 1D bond graph signal is extracted from the hydraulic piston block and sent to the multi solid 3D model. This signal has to be multiplied by a transformer of constant 0.01 since the units that operate the hydraulic system are HeN and cm/s, and it needs to be converted to N and m/s.

To summarise, a flow and pressure are pumped into the system, where a diverter valve divides it between the central and frontal steering systems. In the front steering system an orbitrol valve sends flow to one of the two channels, making the wheels turn by elongating one piston and contracting the other. Each channel is connected to opposite sides of the pistons, so when one elongates the other retracts. For the central articulation the same happens, but in this case a switch valve decides where to send the flow and pressure coming from the diverter valve. This system could emulate the steering of the real tractor, but a control system should be added in order to create automatically the pulses for the switch valve. This control system would track the steering wheel rotation and open the switch valve accordingly.

8 Results

Even though the complex hydraulic steering system provided has not been properly adjusted to work in the model, mainly due to a lack of time, simulations can still be made in the simple one in order to observe the behaviour of the multi solid model. To do so, a set of initial parameters can be set to make it describe a trajectory:

- An initial torque of 200 Nm can be set to the front wheels to emulate the power sent by the transmission. It is a small value, but for a slow speed movement will suffice.
- As shown in the description in the simple hydraulic system, a sinusoidal signal will be sent to the switch vales, making the tractor turn from one side to the other, describing a random trajectory. The angular frequency of the wave will be set to 0.5, so every 6 seconds approximately the switch valves will change the direction of the flow and pressure. This means that every 6 seconds the tractor will turn to one side.

Before starting the simulation, the run parameters have to be set. In this case, by trying different options, the fastest one to make the simulation is the Modified Backward Differentiation Formula, which can be set in the run parameters of 20Sim. With the simple hydraulic system is not necessary, but with the more complex one the integration errors will need to be augmented. If left to 1e-009 the simulation will crash due to requiring too much accuracy.

The simulation is left to run to simulate 25 seconds, and the obtained trajectory is represented in Figure 58.

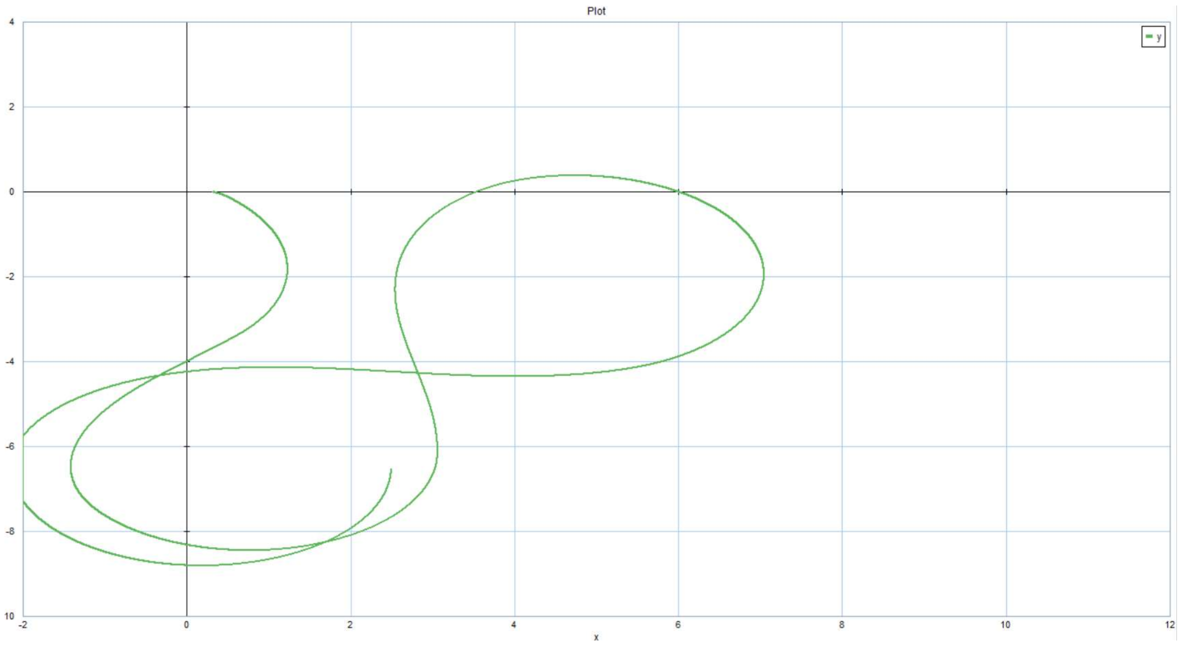


Figure 58. Simulation trajectory

It can be seen how the sinusoidal describes approximately a figure 8, but since a constant torque is applied the vehicle continues accelerating and having more speed, which ends up breaking the eight at the end.

The position of each of the main solids can be observed in the plot in Figure 59 throughout the simulation.

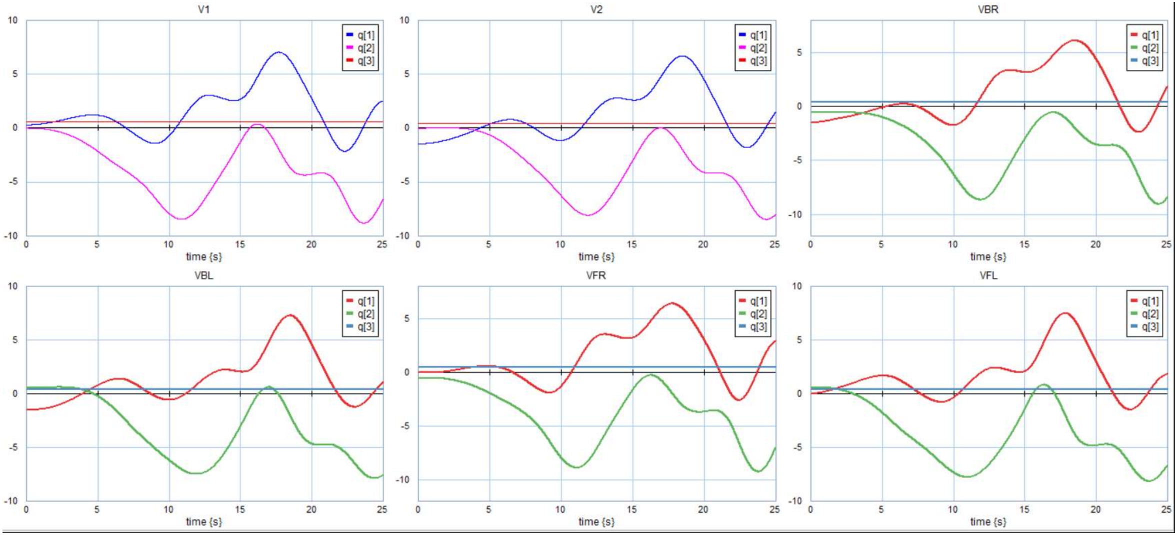


Figure 59. Rigid Solid positions

The elongation of the piston has also been tracked.

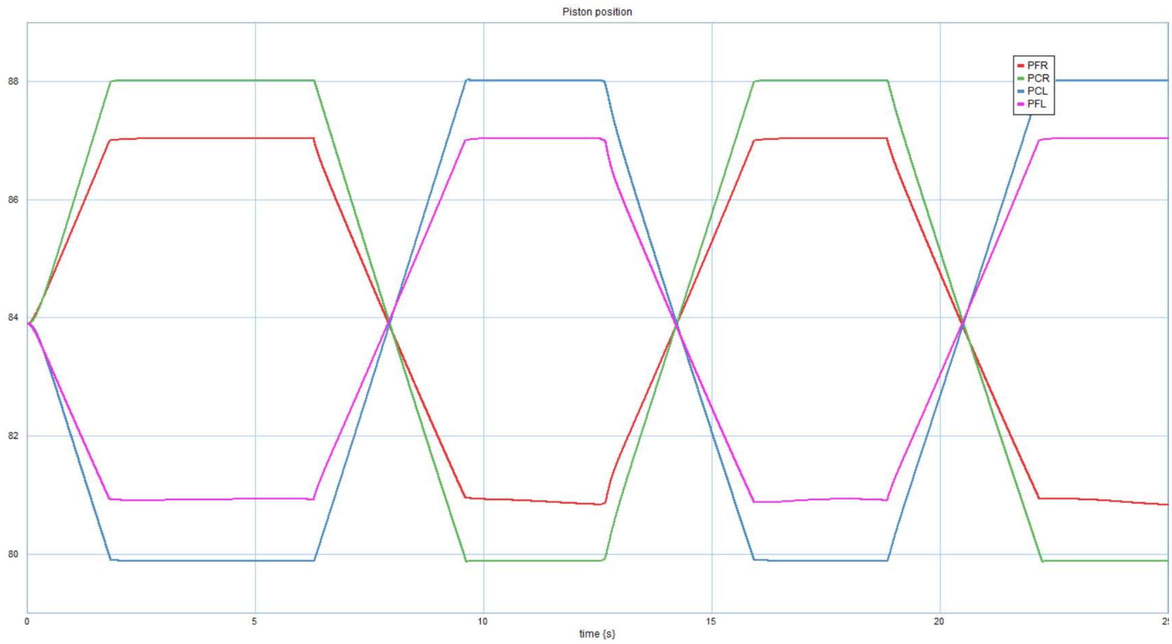


Figure 60. Piston positions

The movement of the piston can be seen in this plot. The sinusoidal function makes the valve switch directions every 6 seconds, and when arriving at their respecting stops, the piston position is blocked until a change in direction is made.

The pressure going to the pistons has also been tracked in Figure 61.

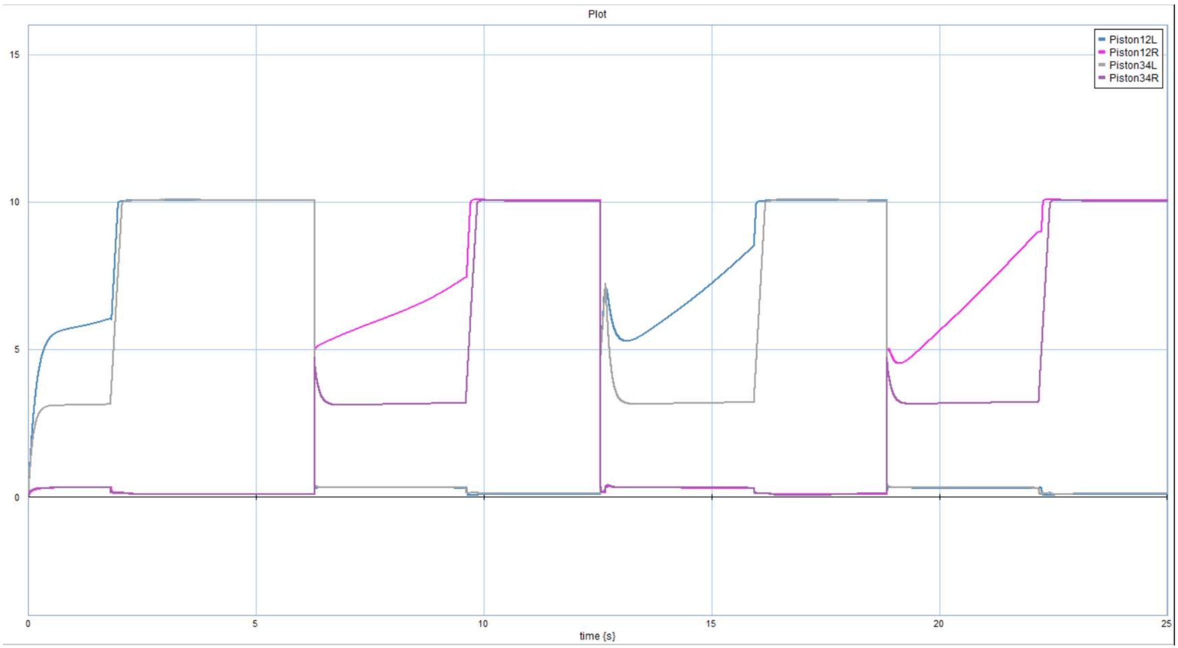
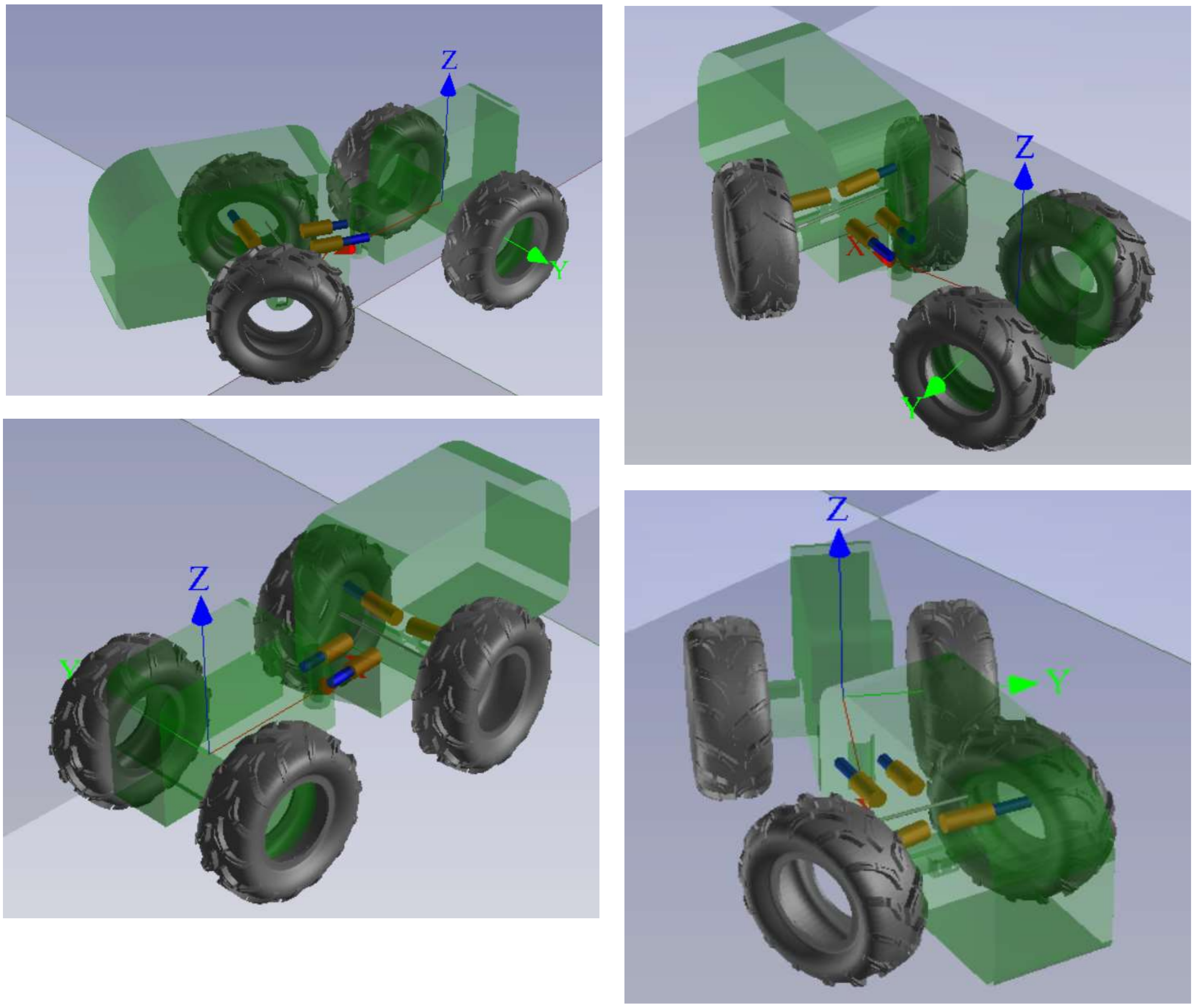


Figure 61. Pressure going to the pistons

Finally, a collage of pictures from the animation is included.



9 Future Works

Even though a working model has been obtained, a list of tasks exists in order to further develop it:

- Incorporation of a real hydraulic system. This task consists in the incorporation of a working bond graph diagram of an approximation of a real hydraulic system that this tractor could incorporate. One of this possible systems has been described in this report, but it still has one limitation. For it to be a real hydraulic system a control should be implemented between the front and central steering. The main difficulty of this task is to achieve a stable simulation, giving the user confidence in the model to simulate complex manoeuvres and trajectories. The model includes a significant number of elements, so it becomes really sensible to the implementation of a complex hydraulic system. This incorporation will be a challenge, but is necessary in order to continue with some of the next tasks.
- Once a real hydraulic system has been incorporated, the next step would be to calibrate the model. Even though the geometry of the model may be more or less accurate, in order to obtain accurate data from the simulation, the dimensions of the vehicle have to be compared to the real ones. This comparison would allow the user of the model to be sure that the set of different points of interest are accurate and describe the correct geometries of the steering systems. This also applies to the masses and inertia moments of the model. This should be obtained by the real CAD design of the vehicle or by experimental methods, but it is basic that these values are the exact same as the real ones. Another important set of dimensions are the travel distance of the piston, which right now is approximately set.
- When it has been established that the dimensions of the model are the real ones, the next step is to calibrate the steering system. The friction of the different steering systems has been established approximately, but these values should be compared to the real ones. To do so, a complete monitoring of the real hydraulic steering system should be done to record the pressures and flows obtained by a given steering input. Once this experimental data is obtained, the friction of the steering system should be set in the hydraulic side of the pistons to emulate the pressures and flows needed for a turn. Once the hydraulic system used in the model is calibrated, meaning it gives approximately the same values obtained experimentally, moving tests can be performed. These tests will have the aim of calibrating the model in order for it to describe the same trajectory as the real vehicle with the same inputs of wheel turning and torque.

10 Environmental and Social Impact

The introduction of modelling tools to the industry enable the reduction of a set of negative factors that affect society and the environment. The creation of multi solid models aids engineers in the test and design phases of a project. Having a model enables the reduction of the number of prototypes required for the commercialization of a product. This helps reduce the energy and resource consumption enabling a decrease in the negative environmental impacts produced during the prototyping phase. If the model is actively used to detect future failure points, a greater optimisation of the life cycle can be done, reducing its overall carbon footprint by fixing issues before important failures occur.

This digitalization of tests helps reduce the exposure to risks of the personnel working in the prototyping phase. It also prevents risks associated to future events of the multi solid systems life, mainly due to accidents. Having a digital model of a machine enables engineers to have a really accurate knowledge of its behaviour, which leads to a better design in the safety area.

11 Economic Impact

Summary of the dedicated time to the project:

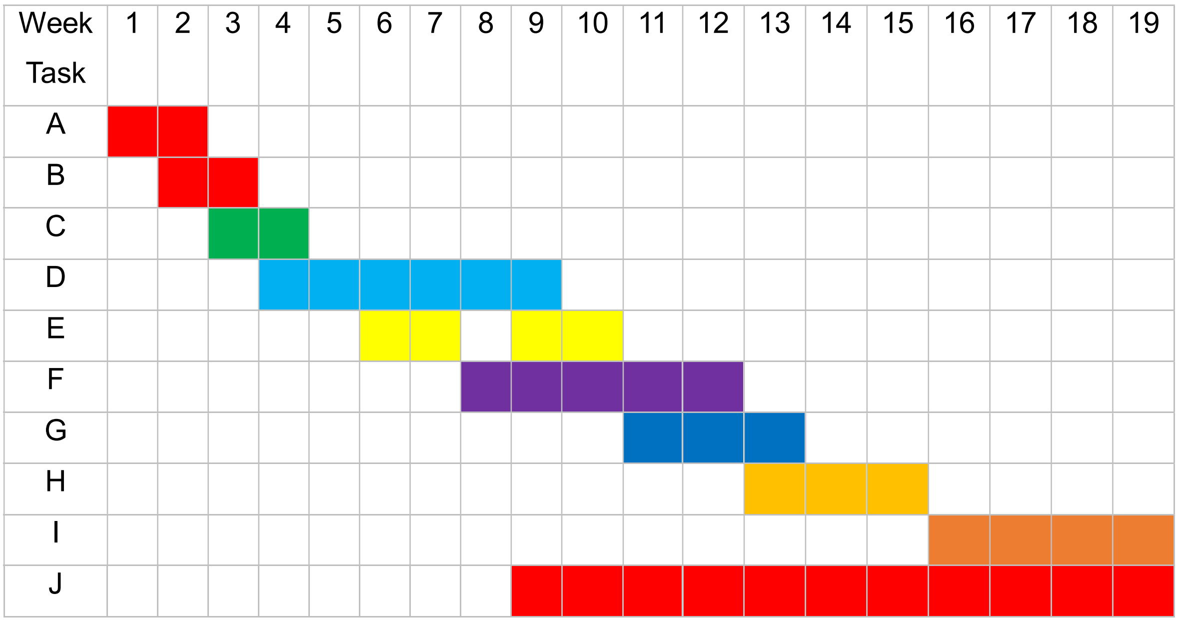


Table 6. Simplified Gandt diagram of the project

The main tasks performed are the following:

- A : Introduction to Multi solid Dynamics and Bond Graph.
- B : Introduction to 20Sim and Bond Graph 3D.
- C : Parameterization of the vehicle (Vehicle description part).
- D : Assembly of the main chassis to the wheels and articulated hubs.
- E : Creation of the animation.
- F : Incorporation of pistons and steering bar.
- G : Incorporation of the Pacejka tire model.
- H : Incorporation of the first hydraulic system.
- I : Incorporation of the real hydraulic system.
- J : Writing of the report.
- K: Meetings

Task	A	B	C	D	E	F	G	H	I	J	K	Total
Hours	18	18	12	50	12	30	24	24	36	56	20	300

Table 7. Hours spent on each task

By an engineer's cost per hour of 20 €, the economic impact of this project would be of 6000€. To this price a future development of 150 to 300 more hours of engineering time would have to be added in order to fulfil the tasks stated in the future work part. This is without including the cost of the experiments needed to do the calibration.

Even though making an accurate model can be expensive, the benefits it provides can lead to a significant reduction of cost in the development and commercialization of the project. Models reduce the prototype needed to reach design goals, and prevent or help detect future problems that machines or vehicles might have. By supporting the already existing experimental methods to test designs with digital models, a reduction in time and cost is reached, facilitating companies to do research and development that previously was outside their reach.

12 Conclusions

Even though the simulation with the complex hydraulic system was not possible, doing it with the simple one has enabled the testing of the multi solid model. With this testing it can be concluded that the multi solid model behaves like a normal articulated tractor would, describing trajectories that would be expected. The simulation produces an animation that significantly aids in the visualization of the behaviour of the tractor, especially both steering systems. The results obtained help in stating that the multi solid model works as intended, and is suitable to accommodate a variety of hydraulic systems to test them. Their incorporation is easily done, with just one connection to each piston, and the parameterization of the model has been reached, with all the geometric parameters within a code block. These parameters are easily modifiable, if the new geometry is correct, and can generate different articulated tractors with one single model.

Bond Graph is an interesting tool that enables the modelling of a big variety of systems, enabling the engineer to deeply understand the models created. The main drawback of simulating with Bond Graph and 20Sim is the sensibility of the simulation to different systems of a given model. The more complex the system becomes, the easier it is for the simulation to crash, so it needs more attention than other software or systems.

Even though 20Sim has demonstrated that the simulation of multi solid systems is possible, it requires more time and effort than other software. Although it provides a better understanding of the simulated model, the added complexity and sensibility of simulations does not make it more appealing than the other commercial software mentioned in this report. Using it for academic purposes would enable students to better understand what the simulation is doing, providing a better learning experience, but its adoption would be harder in the commercial sector.

Due to a lack of time, the approximation of the real hydraulic system hasn't been finished, so it was not possible to correctly incorporate it to the model. There were tests done, but the simulations couldn't be finished correctly. Another issue found in the test and result faze was the lack of testing data to calibrate the model, which led to the lack of a result comparison with real data. This is the main work for the future of the model, a calibration with real data, and the incorporation of different hydraulic circuits, first to do said calibration and then to test them.

In the current state of the model there needs to be more work in making it behave like the real tractor. This include tasks like setting a resistance to the different steering systems so the pressures required to move the pistons are the same as the real ones, right now they are approximations. If the model is needed for a company, or better representation is required, an improved CAD design can be implemented, with all the features and details of the tractor. Even though this will put a heavier load to the computer, for presenting results or for marketing purposes it can be interesting.

As previously stated, if the model is used to simulate a vehicle that moves at higher velocities than the tractor, an interesting modification to work on would be the implementation of the other two efforts in the Pacejka formulas. Since those formulas already appear in this report, their implementation could be easily done following the longitudinal force modelling.

To summarise, the multi solid model has been created, but more work is required in order to achieve simulation results that are comparable to real data. To do so, experimentation with the real tractor would be necessary to do a calibration. On the hydraulic systems side, once the model is calibrated, a set of different circuits will be easily tested on it. And finally, if required, an improvement on the animated model can be done if observing the steering systems behaviour is not required.

13 Bibliography

[1] Urkullu G.(2019). Integración de las ecuaciones de la dinámica de sistemas multicuerpo mediante diferencias centrales de orden dos, Universidad del País Vasco, Bilbao.

[2] Boudon, B., Malburet, F., & Carmona, J. C. (2017). Simulation of a helicopter's main gearbox semiactive suspension with bond graphs. *Multibody System Dynamics*, *40*(4), 375–405.

[3] Ragusila, V., & Emami, M. R. (2014). Modelling of a robotic leg using bond graphs. *Simulation Modelling Practice and Theory*, *40*, 132–143.

[4] *Adams - The Multibody Dynamics Simulation Solution*. Retrieved June 22, 2022, from https://www.mscsoftware.com/product/adams , Irvine California.

[5] *Getting Started using ADAMS*. Retrieved June 22, 2022, from https://www.me.ua.edu/me364/Student_Version/Digital_Appendix/Getting_Started_Usings_ADAMS-Car.pdf

[6] *20-sim | Modeling & Simulation*. Retrieved June 22, 2022, from https://www.20sim.com/ , Enschede Netherlands

[7] *Open Dynamics Engine*. Retrieved June 22, 2022, from https://www.ode.org/

[8] *BCS Agrícola*. Retrieved June 22, 2022, from https://www.bcsagricola.com/

[9] Nagalakshmi, B., Kiranmayi, R., & Nagaraju, R. (2021). Various Domains of Bond Graph. *Turkish Journal of Computer and Mathematics Education*, *12*(2), 772–779.

[10] Heras, S. de las, & Codina Macià, E. (1997). *Modelización de sistemas fluidos mediante bondgraph*. 257.

[11] Xiaotian, L., & Anlin, W. (2014). Definitions of causality in bond graph model for efficient simulation mechanism. *Mechanism and Machine Theory*, *80*, 112–124.

[12] Filippini G. J., Delarmelina, D., Pagano, J., Alianak, J. P., Junco, S., and Nigro N. M., Dynamics of Multibody systems with Bond Graphs, ENIEF, 2007.

[13] Hroncová, D., Gmiterko, A., Frankovský, P., & Dzurišová, E. (2015). Building Elements of Bond Graphs. *Applied Mechanics and Materials*, *816*, 339–348.

[14] Bera, T. K., Samantaray, A. K., & Karmakar, R. (2012). Bond graph modeling of planar prismatic joints. *Mechanism and Machine Theory*, *49*, 2–20.

[15] Carignano, M., Filippini, G., Nigro, N., 2013. Librería Multi-Bond Graphs para Dinámica 3D de Vehículos Automóviles. Presented at the ENIEF MECOM 2013, Mendoza.

[16] Merzouki, R., Ould-Bouamama, B., Djeziri, M. A., & Bouteldja, M. (2007). Modelling and estimation of tire–road longitudinal impact efforts using bond graph approach. *Mechatronics*, *17*(2–3), 93–108.

[17] Zhao, J., Hu, Z., Zhu, B., & Gong, J. (2018). Integrated model control of brake–wheel system using bond graph method. *Advances in Mechanical Engineering*, *10*(7).

[18] *Tyre model*. Retrieved June 22, 2022, from http://www-cdr.stanford.edu/dynamic/bywire/tires.pdf

[19] Drozdz, W., & Pacejka, H. B. (1991). Development and validation of a bond graph handling model of an automobile. *Journal of the Franklin Institute*, *328*(5–6), 941–957.

13.1 Figures

Figure 1. Retrieved June 22, 2022, from http://www.masquemaquina.com/2020/06/sistemas-de-direccion-en-el-tractor.html

Figure 2. Retrieved June 22, 2022, from https://www.noticiasmaquinaria.com/la-cosechadora-palesse-gs3219-de-gomselmash-ahora-en-orugas/

Figure 3. Retrieved June 22, 2022, from https://maquinza.com/diferencia-dumper-articulado-fronta-y-giratorio/

Figure 4. Retrieved June 22, 2022, from https://maquinza.com/alquiler/excavadora-oruga-30-tn/

Figure 5. Retrieved June 22, 2022, from http://multicorpos.com/portfolio-posts/adams-en/

Figure 6. Retrieved June 22, 2022, from https://www.me.ua.edu/me364/Student_Version/Digital_Appendix/Getting_Started_Usings_ADAMS-Car.pdf

Figure 7. Retrieved June 22, 2022, from https://www.20sim.com

Figure 8. Retrieved June 22, 2022, from https://www.20sim.com/webhelp/modeling_tutorial_bond_graphs_bondgraphs.php

Figure 9. Retrieved June 22, 2022, from https://www.tractoresferrari.com/producto/tractor-isodiametrico-vega-85-ar

Figure 10. Retrieved June 22, 2022, from https://www.tractoresferrari.com/producto/tractor-isodiametrico-vega-85-ar

Figure 11. Retrieved June 22, 2022, from https://www.tractoresferrari.com/producto/tractor-isodiametrico-vega-85-ar