UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESIGN AND REAL-SCALE COMPUTATIONAL SIMULATION OF INDUSTRIAL VEHICLES BASED ON MODELS LTM 42043-2 AND LTM 42028-2

AUTHOR: MIGUEL SEGARRA MARTÍNEZ

SUPERVISOR: JOSE LUIS OLIVER HERRERO

Academic year: 2021-22

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

BACHELOR'S DEGREE IN INDUSTRIAL ENGINEERING. BACHELOR'S THESIS

DESIGN AND REAL-SCALE COMPUTATIONAL SIMULATION OF INDUSTRIAL VEHICLES BASED ON MODELS LTM 42043-2 AND LTM 42028-2

DOCUMENT 0.- ABSTRACT & GLOBAL INDEX

AUTHOR: MIGUEL SEGARRA MARTÍNEZ
SUPERVISOR: JOSE LUIS OLIVER HERRERO

Academic year: 2021-22

ABSTRACT

This Bachelor's Thesis consists of the acquisition of the necessary knowledge and skills for the management and mastery of different programs of Computer-Aided Design, CAD, and Computer Aided Engineering, CAE. The use of these tools is to achieve the main objective: to design "self-aligning mechanisms" in a professional way. All work has been done by using "SolidWorks® 2007 SP5.0", which includes the "COSMOSMotion" add-in, "SolidWorks® 2020", and "RecurDyn".

After studying the mechanisms that industrial vehicles and automobiles incorporate, as well as the basis and elements that enable and explain the movement of the mechanisms, it is intended to design all the "constraints" that participate in each of the models, mainly from a kinematic point of view (bearings, …). When said design is made in such a way that the "apparent mobility", or the capacity of movement that a certain mechanism (or mobile system) is intended to display, coincides with the "calculated mobility", or the mechanism's capacity to move according to the arrangement and selection of the constraints, the mechanism is said to be "self-aligning". In space, a body has 6 degrees of freedom (DOF): the rotational movement around each of the three (x-y-z) axes, and the translational movement along these axes. Each 'type' of constraint allows and restricts a certain number of degrees of freedom.

Whenever there be a relative movement between two pieces, the incorporation of some type of constraint should be considered. This is the reason why developing a correct study and design of the mechanisms and bearings of a system is essential. For example, the probability of system failure is reduced, the service life of the mechanisms is lengthened, maintenance cost is reduced, and the system is made as useful as it be designed.

To achieve all this, a tool has been used: the "LEGO® Technic Models" (LTM) with their corresponding commercial code. Once virtualized, they have been called as "vLTm".

Keywords: CAD, CAE, SolidWorks®, RecurDyn, COSMOSMotion, mechanism, self-aligning mechanism, to self-align, degree of freedom, DOF, mobility, bearing, Table of Reshetov, service life, LEGO® Technic Model.

RESUMEN

El presente Proyecto “Trabajo Final de Grado” consiste en la adquisición de los conocimientos y competencias necesarias para el manejo y dominio de los diferentes programas de diseño asistido por ordenador, o *Computational Aided Design* (CAD), y de simulación de ingeniería, o *Computer Aided Engineering* (CAE), todo ello con el principal objetivo de diseñar “mecanismos autoalineados” de forma profesional. Se ha trabajado con el “SolidWorks® 2007 SP5.0”, el cual incluye el *add-in* de “COSMOSMotion”, con el “SolidWorks® 2020” y con “RecurDyn”.

A partir de estudiar los mecanismos que los vehículos industriales y automóviles incorporan, así como de los fundamentos y los elementos que posibilitan y explican el movimiento de los mecanismos, se pretende realizar el diseño de todas las “restricciones cinemáticas” (rodamientos, …) que participan en cada uno de los modelos, principalmente desde el punto de vista de la cinemática. Cuando dicho diseño se realiza de forma que la “movilidad aparente”, o la capacidad de movimiento que se desea que tenga un cierto mecanismo (o sistema móvil), coincide con la “movilidad calculada”, o la capacidad de que se mueva según la disposición y selección de las restricciones, se dice que el mecanismo está “autoalineado”. En el espacio, un cuerpo tiene 6 grados de libertad (gdl): el movimiento rotacional alrededor de cada uno de los tres ejes (x-y-z), y el translacional por dichos ejes. Cada ‘tipo’ de restricción permite y restringe un cierto número de grados de libertad.

Siempre que entre cada dos piezas haya un movimiento relativo, se debe contemplar la incorporación de algún tipo de restricción cinemática. Y es precisamente por este motivo que realizar un correcto estudio y diseño de los mecanismos y rodamientos de un sistema es fundamental. Por ejemplo, se reduce la probabilidad de fallo del sistema, se alarga la vida útil de los mecanismos, se reduce el coste de mantenimiento, y se consigue que el sistema sea tan útil como se diseñe.

Para la consecución de todo ello se ha utilizado una herramienta: los modelos de “LEGO® Technic Model” (LTM) con su correspondiente código comercial. Una vez virtualizados, se les ha denominado como “vLTm”.

Palabras Clave: CAD, CAE, SolidWorks®, RecurDyn, COSMOSMotion, mecanismo, mecanismo autoalineado, autoalinear, grado de libertad, movilidad, rodamiento, cojinete, vida útil, LEGO® Technic Model.

RESUM

El present Projecte "Treball Final de Grau" consisteix en l'adquisició dels coneixements i competències necessàries per al maneig i domini dels diferents programes de disseny assistit per ordinador, o *Computational Aided Design* (CAD), i de simulació d'enginyeria , o *Computer Aided Engineering* (CAE), tot això amb el principal objectiu de dissenyar "mecanismes autoalineats" de manera professional. S'ha treballat amb el "SolidWorks® 2007 SP5.0", que inclou l'*add-in* de "COSMOSMotion", amb el "SolidWorks® 2020" i amb "RecurDyn".

A partir d'estudiar els mecanismes que els vehicles industrials i automòbils incorporen, així com els fonaments i els elements que fan possible i expliquen el moviment dels mecanismes, es pretén realitzar el disseny de totes les "restriccions cinemàtiques" (rodaments, ...) que participen en cada un dels models, principalment des del punt de vista de la cinemàtica. Quan aquest disseny es realitza de manera que la "mobilitat aparent", o la capacitat de moviment que es desitja que tinga un cert mecanisme (o sistema mòbil), coincideix amb la "mobilitat calculada", o la capacitat que es mogui segons la disposició i selecció de les restriccions, es diu que el mecanisme està "autoalineat". A l'espai, un cos té 6 graus de llibertat (gdl): el moviment rotacional al voltant de cada un dels tres eixos (x-y-z), i el moviment translacional per aquests eixos. Cada 'tipus' de restricció permet i restringeix un cert nombre de graus de llibertat.

Sempre que entre cada dues peces hi haja un moviment relatiu, s'ha de contemplar la incorporació d'algun tipus de restricció cinemàtica. I és precisament per aquest motiu que realitzar un correcte estudi i disseny dels mecanismes i rodaments d'un sistema és fonamental. Per exemple, es redueix la probabilitat de fallada de sistema, s'allarga la vida útil dels mecanismes, es redueix el cost de manteniment, i s'aconsegueix que el sistema sigui tan útil com es dissenyi.

Per a la consecució de tot això s'ha utilitzat una eina: els models de "LEGO® Technic Model" (LVT) amb el seu corresponent codi comercial. Una vegada virtualitzats, se'ls ha denominat com "vLTm".

Paraules clau: CAD, CAE, SolidWorks®, RecurDyn, COSMOSMotion, mecanisme, mecanisme autoalineado, autoalinear, grau de llibertat, mobilitat, rodament, coixinet, vida útil, LEGO® Technic Model.

GLOBAL INDEX

LIST OF FIGURES

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

BACHELOR'S DEGREE IN INDUSTRIAL ENGINEERING. BACHELOR'S THESIS

DESIGN AND REAL-SCALE COMPUTATIONAL SIMULATION OF INDUSTRIAL VEHICLES BASED ON MODELS LTM 42043-2 AND LTM 42028-2

DOCUMENT 1.- DESCRIPTIVE MEMORY

AUTHOR: MIGUEL SEGARRA MARTÍNEZ
SUPERVISOR: JOSE LUIS OLIVER HERRERO

Academic year: 2021-22

INDEX 1

LIST OF FIGURES 1

1. OBJECTIVE OF THE THESIS

This Bachelor's Thesis consists of the acquisition of the necessary knowledge and skills for the management and mastery of different programs of Computer-Aided Design, CAD, and Computer Aided Engineering, CAE. The use of these tools is to achieve the main objective: to design "self-aligning mechanisms" in a professional way. It is recommended to read Document 0 > Abstract with the purpose of knowing some brief concepts and details which are related with the objective of this Thesis.

Some kind of kinematic constraints have to be defined between each two pieces with relative movement. The purpose of this project is to use engineering design and simulation tools to understand the relative movements that exist between the different parts of the mechanisms, as well as to self-align them. As it will be shown in Annex 1, the tools used to do all this are highly useful. For example, the fact of using "Lego® Technic" models and "Lego®" pieces is just to determine how an industrial machine works, as well as to determine how mechanically is should be assembled. Through all Document 1, theory about kinematic constraints, how to self-align any mechanism, and the methodology followed to design, self-align and to simulate all the models of the Thesis will be described.

Thus, the **professional approach** is as follows. A client comes to the professional office with a prototype of a machine. The client has created it in a real-size scale just by using Lego Technic bricks, easily accessible through the "Lego's eBay", www.bricklink.com. The person request to determine what type of bearings should be placed between each two pieces. But it must be done in such a way to get a self-aligned machine, because the client wants it to have a useful life as long as possible, and to keep maintenance costs as low as possible. He is against programmed obsolescence.". The engineer accepts the order. Once it is fully finished, the engineer will provide the client with a virtual model of the scaled machine that it has been brought to the office. This virtual model is operative in the SolidWorks® environment (CAD). Secondly, a model in Cosmos Motion or Recurdyn (CAE), with the identification of the type of bearings that it would have to be placed in each joint. Finally, also a kinematic and dynamic simulation of the prototype will have to be delivered to the client. It may allow to improve the prototype from a functional point of view. The simulation could be part of the commercial information of the final product.

In this Thesis, most of the **transversal competences** of the bachelor's degree are practiced. Another objective of writing this Thesis is as follows. *The purpose of "Document #1" is trying to give as maximum information as possible about self-aligning mechanisms (not about how to use the whole CAD and CAE programs) and how to virtually simulate a mechanism. Only physical materialized constraints will be shown in this Thesis. In addition, this Thesis has been created in a schematic, simple and visual way -sort of a manual-. So, the way all this knowledge is going to be shown along all this thesis is in a schematic, complete, visual and direct way.*

2. INTRODUCTION TO MECHANISMS

2.1. FIRST NOTIONS OF MECHANISMS FOR THE THESIS

2.1.1. ABOUT MOBILITY

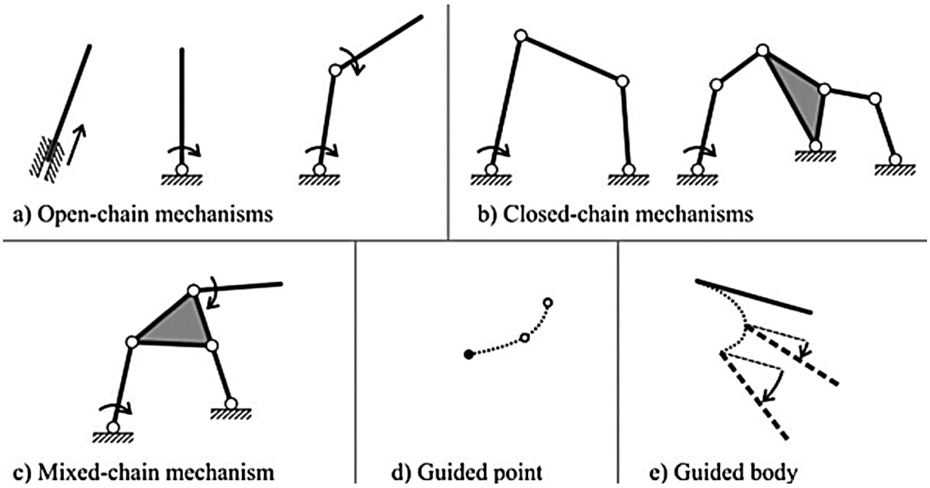
A global system is a group of systems which work as one, and said systems have some characteristics to be analyzed which characterize the global one. In all this Thesis, the considered systems will be mechanisms. Thence, the global system will be the **global mechanism** or **machine** itself.

Some uses of mechanisms are:

- To generate the designed trajectory. For example, rectilinear trajectory generator mechanism
- To change the direction of the flux of power. For example, a worm gear changes the rotational movement by 90 degrees.
- To reproduce a functional action or a needed action for other mechanism

Mobility (M) of a system is the capacity of said system to move. It is measured by its degrees of freedom. In the end, a **mechanism** is just a kinematic chain with at least one piece connected to the piece considered as 'fixed part'. In this topic, a **chain or kinematic chain (KC)** is a group of pieces and constraints within which (1) there is some calculable mobility and (2) there is an identifiable path that, starting from one piece, it is able to go to all the pieces and all the constraints to finally end up in the starting piece. Thus, a **diagram or kinematic diagram (KD)** is the tool which is used to represent a kinematic chain with its calculated mobility, so it will have to be self-aligned.

A classification of kinematic chains can be made as follows:



a) Open-chain mechanisms

b) Closed-chain mechanisms

c) Mixed-chain mechanism

d) Guided point

e) Guided body

Figure 1. Types of kinematic chains. Source: *https://www.cs.cmu.edu/~rapidproto/mechanisms/tablecontents.html*

The concept of kinematic chain lets to study mechanism by mechanism of a machine, as well as the whole machine as a singular mechanism. Because of all said, "mechanism" is going to be always considered as synonym of "considered mechanism"; "chain" is going to be always considered as synonym of “considered kinematic chain”; and all mobilities and the number of redundant constraints (NRC) refers to that considered chain. All this will be better understood after reading all this 2nd section (Doc #1 > Section 2). Thus, before starting to design a mechanism, it is needed to determine what is the wanted objective of the mechanism; how it is going to move; which pieces (parts and assemblies) are going to be involved in getting the wanted mobility, …

The nº of **degrees of freedom (DOF)** represents the number of independent parameters required to specify the position and motion of each body in the system, that is to say, it is the nº of independent movements the considered system can do. In other words, it is the nº of movements (rotation and translation) which need to be stopped/blocked in order to get no output whatever the input may be. Each 'type' of **constraint**, **or kinematic constraint**, allows and restricts a certain number of degrees of freedom. This way, systems are called as “mechanism” if it has >= 1 degrees of freedom, so a dynamic solution is required; “statically determinate structure or static mechanism” if = 0, so it is fully constrained; and “statically indeterminate structure or overconstrained mechanism” if <= -1 -because a negative nº of parameters are needed to block it, what means that it is already stopped-, so there are redundant constraints. This way, mechanisms are actually ‘geometry in motion by equations’ whose behavior is determined by hypothesis, rules, laws, constraints and forces (energy).

For example, when you just fix your elbow, you can rotate your wrist, but you cannot translate your wrist. But when you fix your elbow and you fix your forearm and wrist by squeezing both with your other hand, you cannot rotate the wrist. It is said that a movement has been stopped, or the nº of degrees of freedom has been reduced by one.

In the space there are 6 degrees of freedom: the rotational movement around x, y and z axis; and the translational movement along x, y and z axis. In the plane, there are 3: translation movement along both axles inside the plane, and a rotational movement around the perpendicular axis to the plane. So, a **planar mechanism (2D)** is a mechanism whose bodies’ points move in a plane or in parallel planes; but a **spatial mechanism (3D)** is a mechanism whose bodies’ points move in more than one plane. However, all the parallel planes of the planar mechanism can move at the same time and velocity to keep their parallelism, so that it does not convert into a spatial mechanism. By doing so, the equations to control the mechanism are ‘easier’.

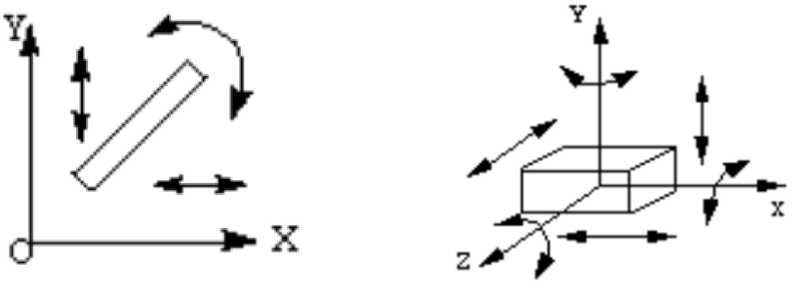


Figure 2. Degrees of freedom in a plane (left) and in the space (right). Source: https://www.cs.cmu.edu/~rapidproto/mechanisms/tablecontents.html

2.1.2. ABOUT PIECES AND THEIR RELATIVE MOVEMENT

In a mechanical system, there are **bodies**, or geometrical figures; **parts**, or each group of bodies (or just one body) which does not have a relative movement between them, but it does have between other parts; **assemblies**, or each group of parts (or just one part) which does not have relative movement between them, but it does have between other assemblies. Usually, the **hypothesis of the rigid solid** is assumed; if not, it is studied from point of view of the **finite elements** brunch of science

In practical terms, ‘body’, ‘part’ or ‘assembly’ can be used to express a **“piece”** of the considered mechanism. However, the purpose of making that differentiation is because CAD and CAE programs cannot create relative movement between two pieces located in the same file. To let there be relative movement between two pieces, it is needed to define each piece in a different CAD file. Thus, it is possible to define **mates**, or geometrical relations between two pieces defined in the CAD file. When the CAD file is finished, it can be exported to .x_b. In the CAE program, the .x_b must be imported, and the user can start to define joints. A **joint** is a type of kinematic constraint between two pieces and it is established to let there be relative movement between them (not between a piece and the inertial frame). When a mechanism has the same behavior with different constraints, both designs are **kinematically equivalent**.

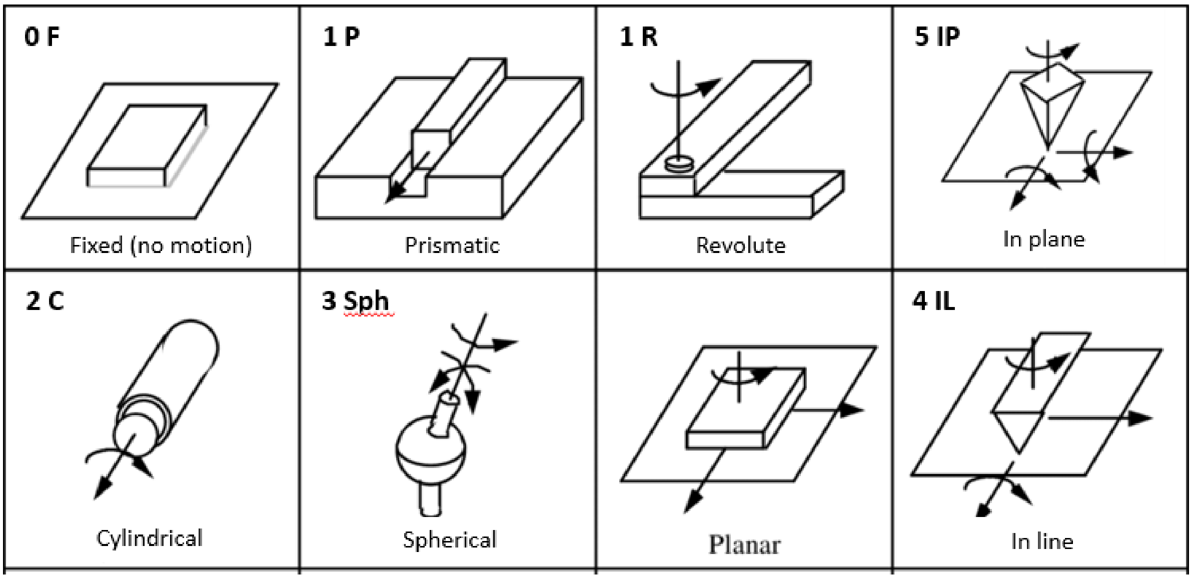


Figure 3. Visual examples of common joints in the Table of Reshetov. Next to its letter, there is a nº which indicates the nº of its degrees of motion (arrows). Source: “Incorporating geometric feature variation with kinematic tolerance analysis of 3d assemblies”; Dabling, Jeffrey.

The equations used by CAE programs to calculate the mobility will appear next in this Thesis. Because of that, only non-redundant constraints must be defined between each two pieces in the CAE program if needed. Once some requisites are achieved, simulation process can start. What is “redundant” will be explained after the “Chebychev–Grübler–Kutzbach criterion” in Doc #1 > Section 2.1.3.

Each mechanism is a group of parts and assemblies which move to respect something. So, it is needed to consider one body as fixed. There are two cases, but in both a joint fixed (F) is created in the program: (1) it does not move with respect to the **inertial frame or ground**, so it is **fixed to the ground**, what is represented by some small diagonal lines; (2) it cannot move with respect to some other piece/s from the considered mechanism, so it is **fixed**. Thus, only one piece is needed to be considered as fixed to the ground, because any other piece can be fixed to this one. The remaining pieces are considered as **moving parts** with respect to fixed to the ground ones

As it has been said, movement is defined between two pieces. In consequence, it is needed to define constraints between two pieces to let there be a relative movement between both. By doing so, the design of the kinematics of the mechanism is taking place. There are different types of constraints: **joints, contact, couplers,** and **motion**. All constraints are defined in order to make the mechanism behave as wanted. Each constraint type lets a specific movement and deletes at least one degree of freedom. For example, between a wheel and the floor it could be defined a constraint "3D contact"; between two gears it can be defined a coupler; or between the mirror and the door's chassis can be defined a revolute joint to let the mirror to move with respect to the door's chassis. All about physical materialized constraints will be explained through this Doc #1, but just highlight **bearings** are a kind of joint according to the **Table of Reshetov** (in Doc #1 > Section 4.1.1.1).

Also **forces** can be defined to study the dynamics of the mechanism. Once all these inputs are defined, the mechanism can be **simulated.** Even, ***self-balanced machines*** can be designed and built. These machines are designed in a way that they operate because of their own forces (and the gravity, of course). So they can need very little energy to operate but being useful. For example, "Ten Fold Engineering" business company does it.

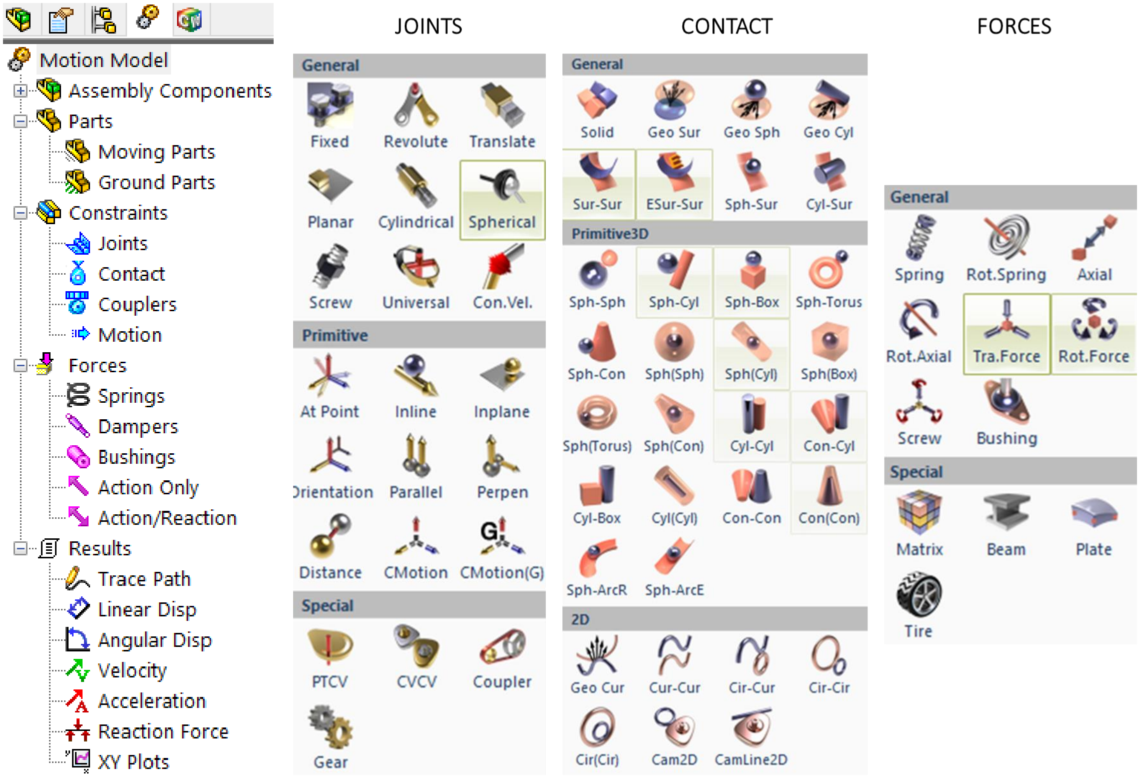


Figure 4. An abstract of pieces and their relative movement through a CAE program display. Left: COSMOSMotion global panel. Right: all types of joints, contacts, and forces available in RecurDyn.

2.1.3. MOBILITY THROUGH EQUATIONS AND DIAGRAMS

The way a constraint is virtually created is through equations. Obviously, they try to represent the real behavior. Thus, every constraint is actually a force or a momentum. They constrain lineal or angular movements. Obviously, 6 = nº removed DOF + nº possible DOF by the joint. In the Table of Reshetov (in Doc #1 > Section 4.1):

- the total nº of these **reaction forces and reaction momentums**, or nº of removed DOF by the bearing, is what appears as “clase” (class in Spanish).
- the nº of possible movements by the joint, or connectivity of the joint, is what appears as “movilidad” (mobility in Spanish).

It has already been said that a mechanism is a group of bodies which satisfy some conditions and functional purposes. In both requirements, the common concept is the mobility (M) of the machine, and so the mobility of all its mechanisms. Each one leads to the justification of building the machine, and the machine’s behavior, properties, characteristics, and functionality. So, the way a mechanism moves is called **apparent mobility**. In other words, it is the degrees of freedom the machine has.

However, since all mechanical parts obey natural laws and these can be expressed as mathematical laws, it is needed to enter constraints into the computer file. By doing so, a **calculated mobility** will appear. It can be calculated from a 2D or a 3D point of view, although at the end all bodies are three dimensional. So, the constraints in a 2D and 3D design do not have to be the same. Obviously, apparent 2D and 3D mobilities must be same; and to calculate the two-dimensional mobility of a spatial mechanism does not make sense.

In addition, it does not sound crazy to think that it is needed to make coincide the apparent mobility (what is wanted) with the calculated mobility (what is designed, how the mechanism works). When this is done, it is said that the mechanism is a ***self-aligning mechanism***. So, the **number of redundant constraints (NRC)** is calculated as some mobility minus other mobility: $NRC_{calc} = M_{ap} – M_{calc} = 0$, where ‘calc’ is ‘2D’ or ‘3D’; so, an auxiliar equation can be defined: only for planar mechanism: $NRC_{23} = M_{2D} – M_{3D}$. Thus, the value of NRC gives information:

- If = 0, it is already self-aligned. To properly self-align a machine, all its mechanisms must be self-aligning.
- If < 0, it gives the number of DOF that must be reduced, what means it has **internal mobility (M_{int})**: $M_{int} > 0$. To have internal mobility is not wanted because there would be uncontrollable movement between at least two pieces. Even using a specific geometrical configuration (for example, axis of rotation perpendicular to a plane, …) would not be enough to make sure the mechanism would work properly whatever it happens.
- If > 0, it gives the number of DOF that must be augmented. $M_{int} < 0$

Using some tool to visualize how the self-aligning process is going on, but even more how it has already finished, can be very useful. This tool is known as kinematic diagram. It is a drawing where all the pieces, constraints and mobilities of the considered mechanism appear. Next, it will be shown how to do it. It can be analyzed in two dimensions if it is a planar mechanism, so M_{2D} is calculated; or in three dimensions if it is a spatial mechanism, so M_{3D} is calculated. In any considered mechanism, at least one piece has to be considered as fixed to the ground.

During all this process of self-aligning a mechanism, mobility can be calculated by using the "**Chebychev–Grübler–Kutzbach criterion**": M_{3D} = 6 * nº total of bodies – the sum of all non-redundant constraints' mobility = 6 * nº of moving bodies – 6 * nº of fixed to the ground bodies – summatory of (6 – nº DOF blocked by that constraint) for all constraints. So, (6 – DOF blocked by that constraint) = f_i is the mobility/connectivity of the constraint i-th; and (nº of moving bodies + 1 body fixed to the ground) = Nb is the total nº of bodies. So, M_{3D} = 6 *(Nb – Nco – 1) +summatory of f_i for all constraints. And so, if 6 is changed by 3, M_{2D} is calculated.

However, it is very important to highlight that to properly use this criterion, it is needed to <u>only take into account the non-redundant constraints</u>. Obviously, when self-aligning a mechanism in a program which uses this criterion, not to create redundant constraints must be considered. For example, if an axis only needs a revolute joint but there are two supports, only is needed to create one revolute joint in one of the supports. To sum up:

$$\mathbf{M_{calc} = M_{calc}\,(\lambda - f\_i) = M_{calc}\,(f\_i)}$$

$$\mathbf{M_{calc} = (\lambda - f\_i) = \lambda * Nb - \sum_{i=1}^{i=Nco} (\lambda - f\_i)}$$

$$\mathbf{M_{calc}\,(f\_i) = \lambda * (Nb - Nco - 1) + \sum_{i=1}^{i=Nco} f\_i}$$

- M_{calc} := mobility of the considered kinematic chain of the considered mechanism. It indicates <u>the nº</u> of possible movements (DOF) it has.
    - "calc" := identifies if it is a planar mechanism ("2D") or a spatial one ("3D").
    - λ = {3 if M_{2D}; 6 if M_{3D}}.
- Nb := nº of total bodies = Nfgb + nº of moving bodies.
    - Nfgb := nº of fixed to the ground bodies = 1. Its identification is usually 1 in the kinematic diagram. The ground or inertial frame is not considered as a body.
- Nco := nº of total <u>non-redundant constraints</u>.
    - i := i-th non-redundant constraint created.
- f_i := mobility/connectivity of joint i-th, that is to say, it is the sum of the nº of movements that joint i-th lets by itself (freedom).
- 6 – f_i := class of joint i-th, that is to say, it is the sum of the nº of movements that joint i-th blocks by itself

$$\mathbf{M_{int} = -\,NRC_{calc}}$$

$$\mathbf{NRC_{calc} = M_{ap} - M_{calc}}$$

$$\mathbf{NRC_{23} = M_{2D} - M_{3D}}$$

- M_{int} := internal mobility
- NRC_{calc} := nº of redundant constraint (NRC) in a 'calc' mechanism. If:
    - $NRC_{calc} > 0$ is the nº of DOF to be augmented.
    - $NRC_{calc} < 0$ is the nº of DOF to be reduced.

  - ○ $NRC_{calc} = 0$ is the condition of self-aligning. Thus, $M_{ap} = M_{calc}$ when it is self-aligned. However, it is needed to check and to interpretate if geometry affects to the output of the equations.
- NRC_{23} := auxiliar equation to calculate NRC for 2D mechanisms. If self-aligned, $NRC_{23} = 0$.

For example, a joint revolute (R) just lets a rotation movement, so its f_i = 1 (rotation Z), because it has blocked 6 – 1 =5 movements (translation X-Y-Z and rotation X-Y).

However, in rare instances, only using previous equations can lead to misleading results.

1. For example, in following figure, equations give $M_{2D} = 0$ but it can move because of geometry. If all pivoted links are the same size, and the distance between the joints on the fixed frame and coupler are identical, it can move with one DOF. The center link is redundant, and because it is identical in length to the other two links attached to the frame, it does not alter the action of the linkage. There are other examples of mechanisms that violate the Gruebler's equation because of unique geometry. A designer must be aware that the mobility equation can, at times, lead to inconsistencies.

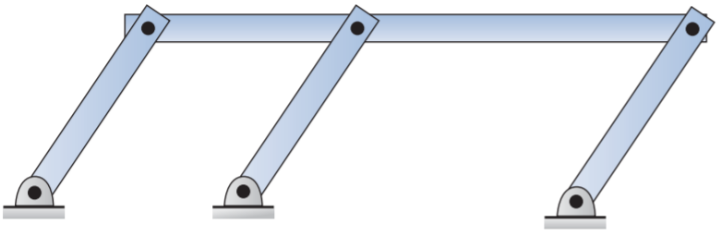


Figure 5. Mechanism that violates the Gruebler's equation. Source: "Machines and mechanisms. Applied kinematic analysis", David H Myszka.

When only prismatic joints create a kinematic chain, +1 piece and +1 revolute (R) joint must be considered (see Doc #1 Section 5.1)

Now it is time to explain what "**redundant**" implies. To do so, compliant bodies are going to be used. Next figures are created by Professor Jonathan Hopkins, when explaining "Design of Architected Materials Using Freedom and Constraint Topologies (FACT)".

Redundantly constrained systems are said to be over-constrained systems. It is the case of the next 3rd figure. Here, 3rd wire does the same as the other two wires in the system. In other words, if 3rd wires is changed or subtracted, it does not affect system's kinematics nor changes its DOF.

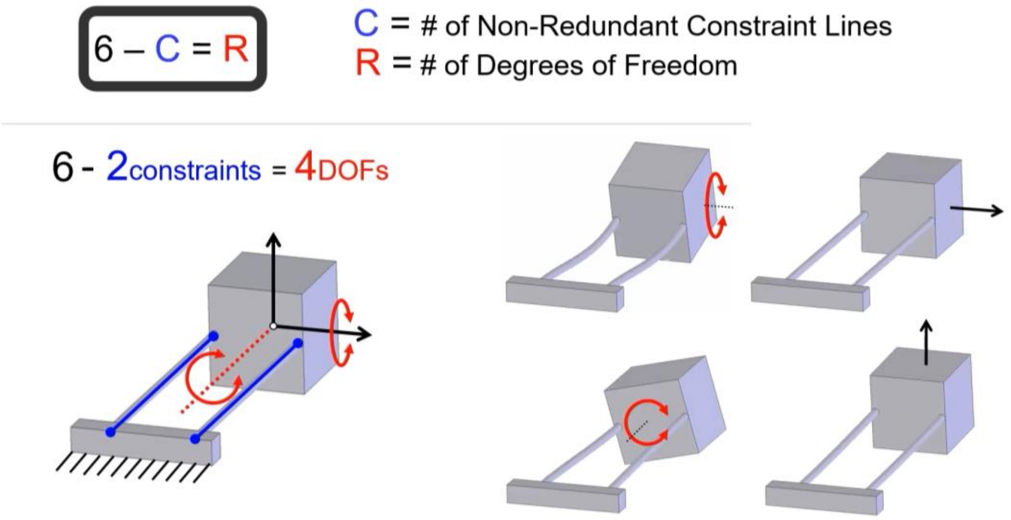


Figure 6. A system without redundant constraints and 4 DOF. 6 – 2 = 4 DOF is okey. Source: "[Freedom And Constraint Topologies]" (FACT); Prof. Jonathan Hopkins at UCLA.

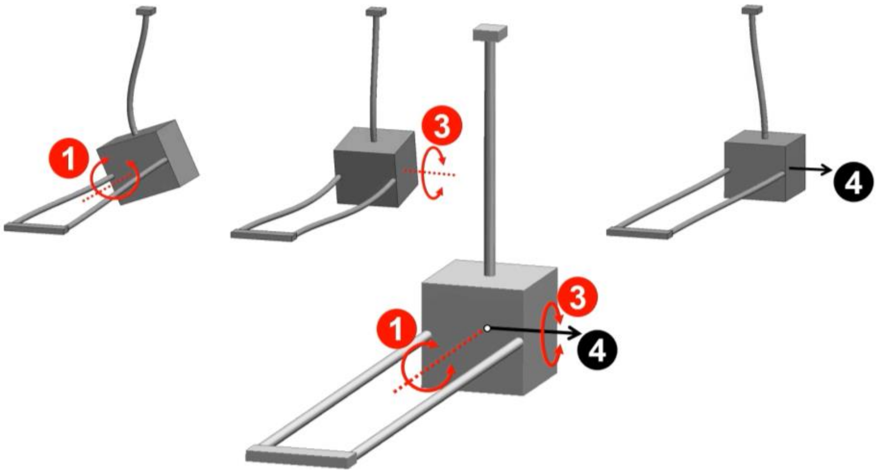


Figure 7. A system without redundant constraints and 3 DOF. 6 – 3 = 3 DOF is okey (1/2). Source: "[Freedom And Constraint Topologies]" (FACT); Prof. Jonathan Hopkins at UCLA.

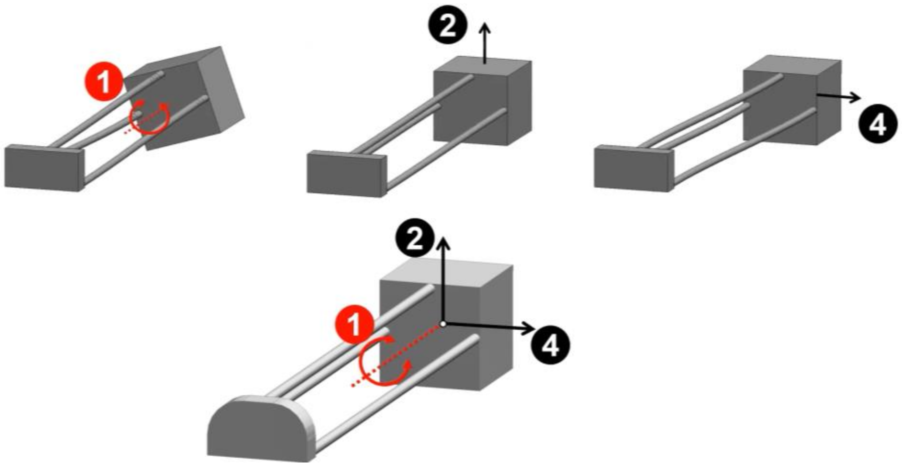


Figure 8. A system without redundant constraints and 3 DOF. 6 – 3 = 3 DOF is okey (2/2). Source: "[Freedom And Constraint Topologies]" (FACT); Prof. Jonathan Hopkins at UCLA.

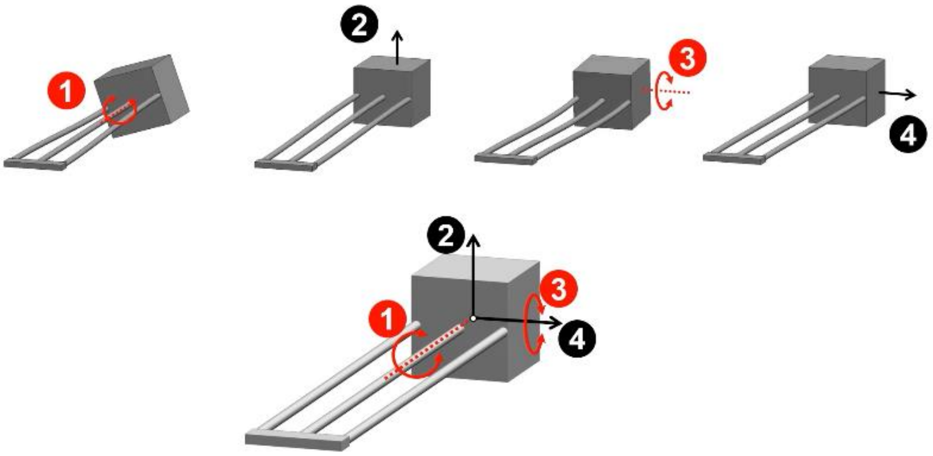


Figure 9. A system with redundant constraints but 4 DOF. 6 – 4 = 2 DOF is not okey. 6 – 2 non-redundant constraints = 4 does be okey. Over-constrained system. Source: “Freedom And Constraint Topologies” (FACT); Prof. Jonathan Hopkins at UCLA.

Also, pieces can be redundant. If both pieces are redundant, it might be needed to consider one fixed joint between them, so they become as only one piece for the CAE program.

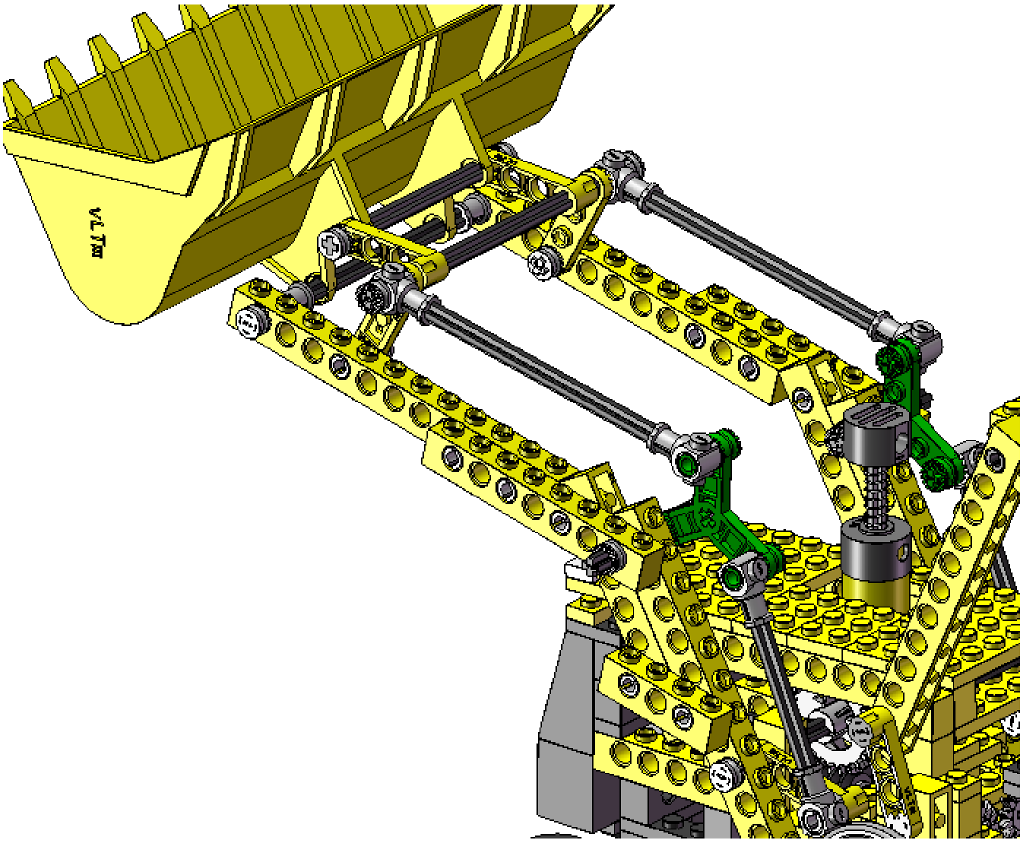


Figure 10. Both green pieces are redundant pieces. To get the LTM_8862’s mechanism “frontal shovel” self-aligned, a fixed joint between both might have to be defined. This way, it becomes as only one piece.

Thus, to self-align a mechanism can be materially achieved by:

- Changing geometry and/or pieces (a mechanism is geometry in motion).
- Changing the type of bearings once the assembly has been designed.

The consequences of this characteristic in mechanism are the following:

- Huge increase in the possibility that expected results are the same as what happens.
- Reduction of maintenance costs and probability of failure during service life. Reduction of loss because of a failure.
- The only inconvenient to get a self-aligned mechanism is that it might be needed higher knowledge to do a proper design and a self-aligning process of a machine, as well as an increment of initial investment because of required mechanical objects(bearings).
- Increment of guaranteed period. Decrement of purchases; thrown away materials (energy and resources from the Earth); resources consumed in the life cycle of the product, from its designing until its recycling. Against programmed obsolescence.
- Service life increases with respect to the same work conditions

  The reason of this is because as bearings get deteriorated, the own relative movement of both pieces joined by it facilitates that the bearing keeps getting shaped. But since it is self-aligning, the shape will make it become as needed to keep having the apparent and calculated mobility equal.

  For example, continuing with the example given when explaining the DOF, imagine the second hand you use to block your wrist is a bearing. The more it be squeezed, the less it can be moved, the hotter surfaces get, the more energy you lose, the more erosion appears, the less efficient is your output movement… However, if instead of having such a restricted movement (a restricted kinematic configuration because of some nº of redundant constraints NRC >0) it were a bearing such that you can move properly without difficulties, that mechanism would not have excessive constraints nor problems.

- Increase of security and result's quality.

  Because of previous point, a bearing will not get blocked nor will change its smoothness.

  For example, when a CNC machine needs less than ten micrometers precision, roughness matters. Or, for example, when a door is installed with three 'revolute' joints (bearings type 'revolute'), it will end up failing or creaking because it cannot be a self-aligning mechanism. There are 1 fixed to the ground piece, 1 piece (the door itself) and 3 joint revolute (R): $M_{2D} = 3*(2-3-1) + 1R*3 = -3$ DOF

2.2. GENERAL ASPECTS ABOUT MECHANISMS

Relative movement is the movement of one piece with respect to another one. For example, when two cars crash when they are going in opposite directions, they experience the sum of both velocities. But if they were going in the same direction, they just would experience the difference between both velocities. The reason is because angular and linear velocity is a vector.

The power flows from the input piece, going through constraints and pieces, and it finally exits through the output piece.

- The **input piece** or **driver** or **driving piece** is a piece whose movement does not come from another piece of the mechanism but from an external to the considered mechanism power supply.
- The pieces between input and output pieces are called **connecting links** or **coupler pieces** or **idler**.
- The **output piece** or **follower** or **driven piece** is a piece whose movement does come from another piece of the considered mechanism, and it either connected to another piece (the kinematic chain is closed) or not connected to anything else (the kinematic chain is opened).
-

The mechanical advantage of a mechanism is the ratio of the output power (or force or torque) divided by the input force (or force or torque). In a linkage, the transmission angle quantifies the force transmission through a linkage and directly affects the mechanical efficiency. Clearly, the definitions of transmission angle depend on the choice of driving link

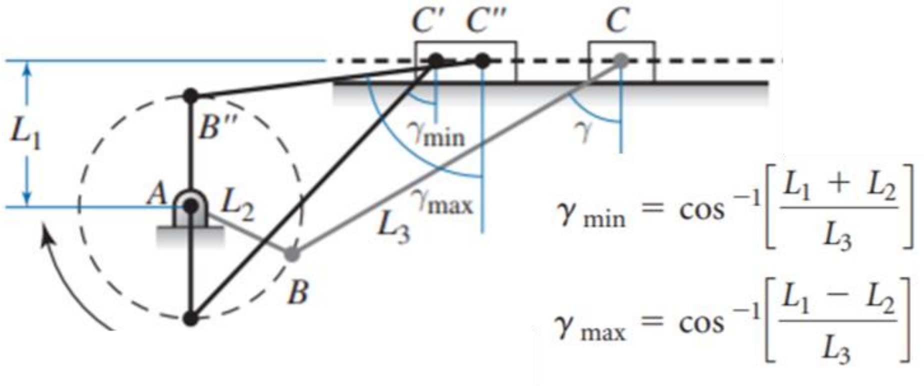


Figure 11. The mechanical advantage of a mechanism depending on power flow and angles. Source: "Machines and mechanisms. Applied kinematic analysis", David H Myszka.

Extreme position is a kinematical configuration of the pieces which implies that it is not sure how the mechanism is going to be behave. It depends on the input piece. It can break, or get blocked and break, or change its kinematic configuration, or get an unknown kinematic configuration… They are commonly reached when an alignment between pieces happens, and said alignment usually is because pieces get extended to their maximum or because they are coincident to other.

For example, in the next figure, if the input is the pink piece L2, it will be considered an extreme configuration. But if it does in blue piece L1, it will not.

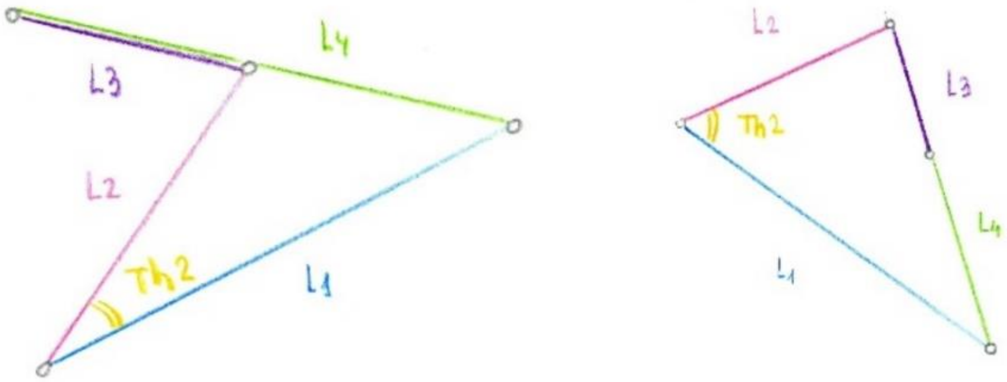


Figure 12. Some 'types' of extreme positions. Left: by coincidence. Right: by extension. Source: *http://www.upv.es/vltmodels/index.html*

Here there are two examples of kinematic deconfiguration. The trajectory of the mechanism changes depending on the initial configuration:

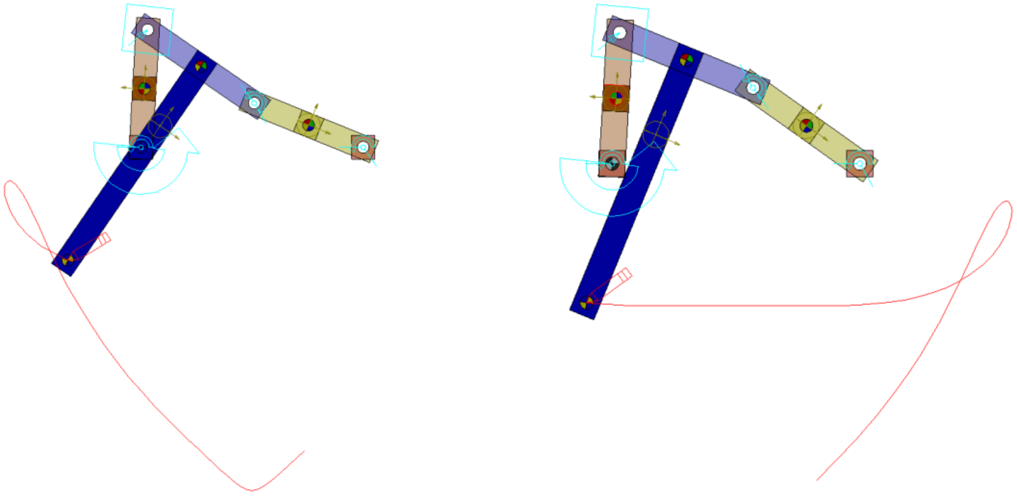


Figure 13. Kinematic deconfiguration. Trajectory of the mechanism depending on the initial configuration.

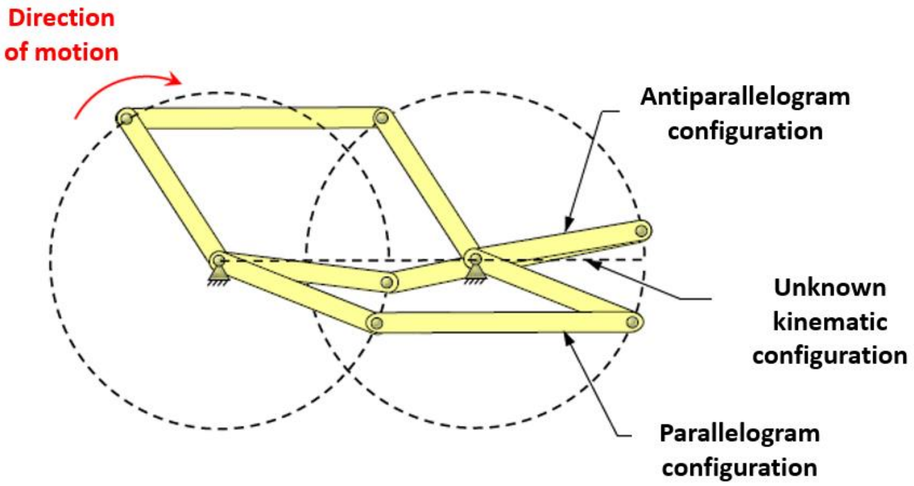


Figure 14. Kinematic deconfiguration in a four-bar linkage. Source: notes of the subject "Theory of mechanisms" in the Bachelor's degree in Industrial Engineering at the UPV.

Since a machine is not a structure, all about its movement must be considered: rotation, translation, motion, trajectories, generalized trajectories, range of motion, circular motion, rectilinear motion, parabolic motion, functionalities of each degree of freedom; cyclical movement (cycle, period, phases, …), continuous movement, alternative movement, intermittent movement; forces, acceleration, drag force … The purpose of building anything is its functionality. In this case, a main advantage of mechanism is they can move. So, since they have some degree of freedom to move, it is needed to define all movements between pieces. Thus, it can be found that a piece can have free motion or predefined motion. First case is related with a free movement of a piece: the piece is not fully materially connected to another one, so its trajectory depends on external forces applied on it, where those forces do not come from any other piece which is mated with. Second case has relation with a **desmodromic movement of a piece**: the piece is fully materially connected to another one, so its trajectory does not depend on external forces applied on it, where those forces come from any other piece which is mated with. In machines, desmodromic movement is wanted for all pieces which are not easily accessible or are hidden or do not help to do any needed function.

For example, in the right following figure, the valve is closed and opened because of direct contact with the no concentric axis. In the CAD program, that is modelled by adding a mate cam between both pieces. This way, the movement of closing and opening the valve only depends on the movement of the axle. It cannot move other way, and its movement just depends on the axle movement. On the other hand, in the left valve, for example the string could get contracted, and so the valve, but not because of the axle movement. Thus, the axle is not fully defining all possible movements of the valve. Left valve have free movement; right valve does not.

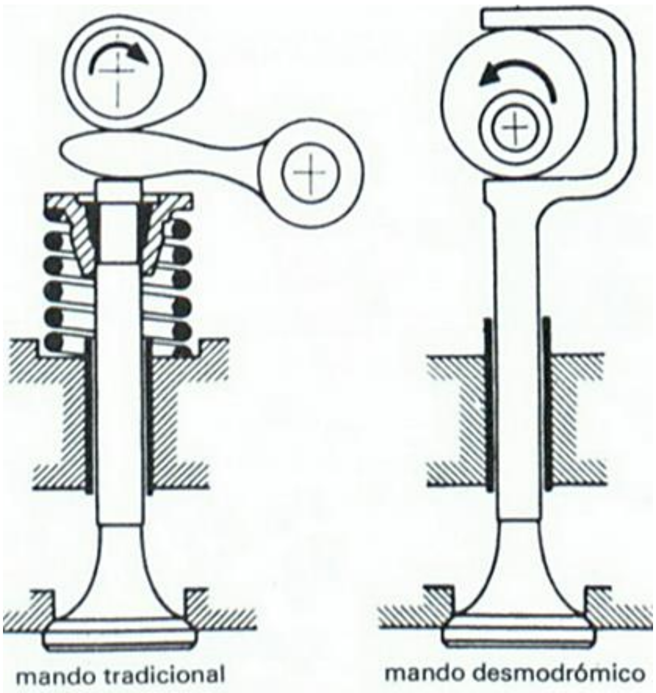


Figure 15. Free movement (left) VS Desmodromic movement (right) of a piece. Source: https://diccionario.motorgiga.com/diccionario/desmodromico-mando-definicion-significado/gmx-niv15-con193829.htm

A **gear** is a toothed wheel. Two pieces with contact surfaces can create a mate gear. This mate let the direction of the output movement be different to the direction of the input movement. In addition, it lets to have a different torque (different force), and if so a different velocity because the power approximately keeps constant: **P = constant = F * v = M * w, where v = w* R because it is a circular motion**. There are very different types of possible gears (circular, no-circular, with no-constant transmission factor, …). Also, depending on the arrangement, they have some properties and called by a specific name.

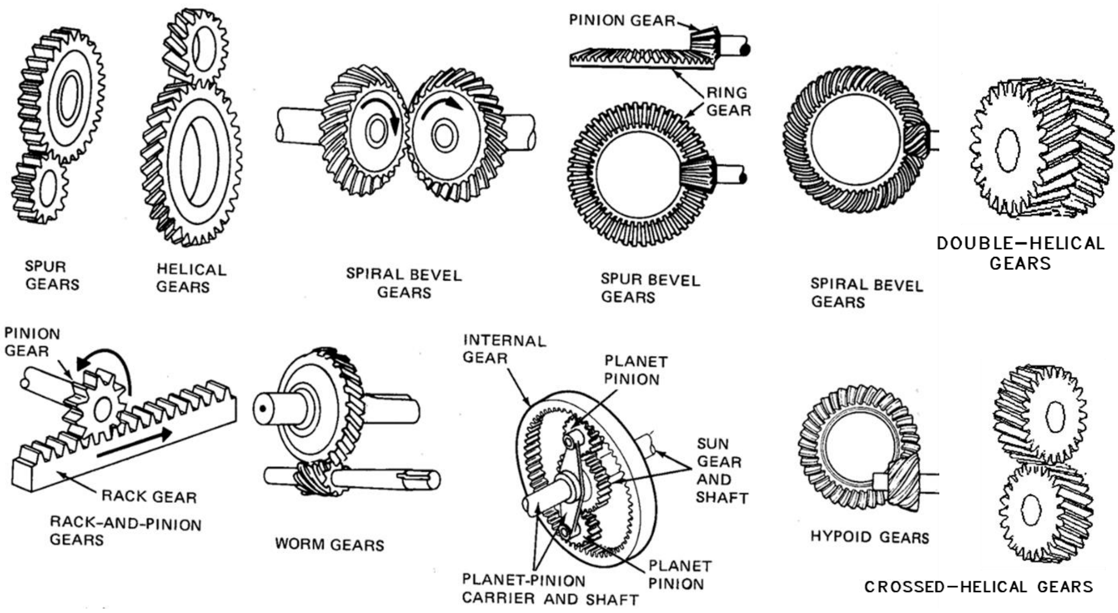


Figure 16. Some types of gear systems.

A worm gear is a gear consisting of a shaft with a spiral thread that engages with and drives a toothed wheel (a gear). Worm gears are a version of one of the six simple machines. Despite the fact lubricant is needed because of relative movement of contact surfaces, the first reason to use a worm gear is the high reduction ratio. A worm gear can have a massive reduction ratio with little effort. Thus, it can be used to greatly increase torque by reducing rotational velocity. It will typically take multiple reductions of a conventional gearset to achieve the same reduction level of a single worm gear, meaning that worm gears reduce nº of moving parts and of places for failure. A second reason to use a worm gear is the inability to reverse the direction of power. If it happens, the mechanism blocks without external energy. Because of the friction between the worm and the wheel, it is virtually impossible for a wheel with force applied to it to start the worm moving

As it has already been said, in almost all mechanisms there is usually a piece with which the movement begins and another in charge of executing the desired task. Such parts are given a variety of names in Engineering. Next, they are going to be referred as driver and follower. From a mathematical point of view, they can be considered as the **independent variable** and the dependent variable in a mathematical function. In this way, in computing it is understood as input and output.

An essential parameter in the analysis of many mechanisms is the ratio of a rotational movement (the angle turned, …) or a translational movement (distance traveled, …) by the follower to the corresponding magnitude of the driver. The quotient between the magnitude of the follower and the driver, in that order, is called the quotient or **transmission factor (TF)**. The transmission factor is a number with sign, to consider the direction of rotation. It constitutes a geometric characteristic of the mechanism and is determined by the shapes and sizes of the parts that compose it. The inverse of this number is called **mechanical advantage (MA)**.

In the case that the considered mechanism is a gear, the variable to be analyzed is the angular velocity of the gears; and the TF is also called the "multiplication ratio" (or reduction), and the inverse of this number is known as the **gear ratio (GR)**. Thus, TF of gears is the ratio of angular velocity (w) of the output gear divided by the angular velocity of the input gear. If the absolute value of TF is >=1, it is called mechanical disadvantage; if <1, mechanical advantage. Regarding the sign, if the ratio is >0, it is because both gears rotate in the same direction; if <0, they rotate in the opposite direction; if =1, it because either both gears have the same angular velocities or because they have no relative movement (for example when both gears are fixed to the same axis). To determine the direction, either "right hand rule" or "right hand thumb/grip rule" can be used.

Wheels/Gears 1 and 2 need to have the same module (m) to rotate so m_1 = m_2 , where m is the diameter of the gear (D) divided by the number of gear's teeth (Z): m = D / Z. They also need to have the same velocity (v) in the contact point so v_1 = v_2. In addition, both wheels rotates so the contact point's velocity is proportional to its radius: v = w * R, where R = D/2. Because of these three conditions, GR can be calculated either with diameters or with number of teeth of the gears:

$$\mathbf{GR = 1 \,/\, TF}$$

$$\mathbf{TF = w_{out} \,/\, w_{in} = D_{in} \,/\, D_{out} = Z_{in} \,/\, Z_{out}}$$

In TF(A to B) = TF(AB), when A rotates 1 turn, B rotates TF(AB) turns. Thus,

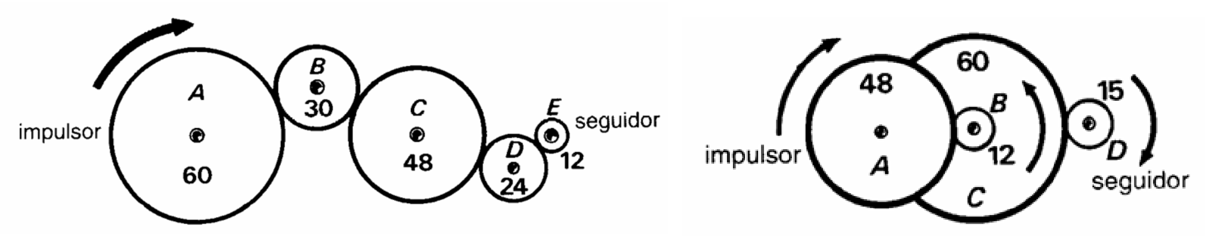


*Figure 17. Simple gear trains (left). TF(AE) = TF(AB)*TF(BC)*TF(CD)*TF(DE) = (- 60/30)*(- 30/48)*(-48/24)*(-24/12) = + 60/12 = +5. Compound gear trains (right): TF(AD) = TF(AB)* TF(BC)* TF(CD) = (-48/12)*(+ 1)*(- 60/15) = + 16.*

Furthermore, **in planetary gears, Willis' formula** is a needed equation to analytically resolve them. It can be used to solve by equaling to another TF equation. In a planetary gear there is one sun gear, at least one planet carrier or arm, one ring gear and one planet gear. The power flow can go in any direction, but it must satisfy next equation:

$$\mathbf{TF = \frac{w_{out} - w_{arm}}{w_{in} - w_{arm}}}$$

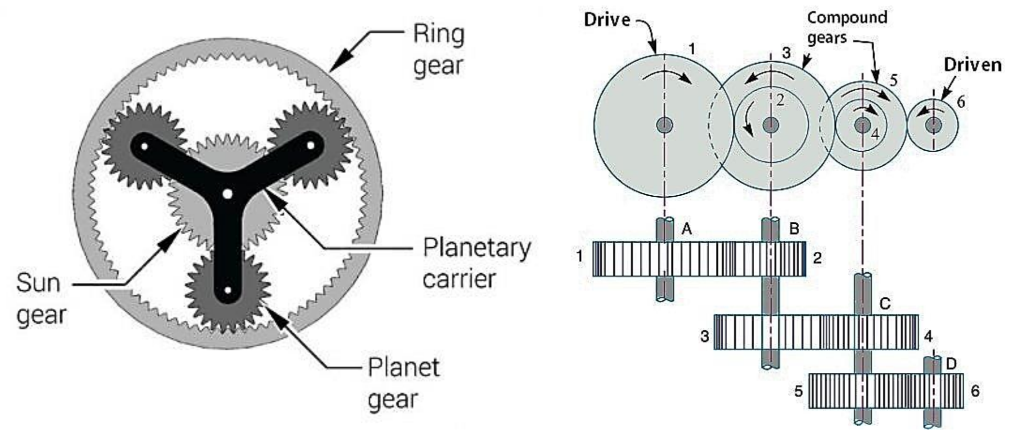


Figure 18. Elements of a planetary gear system (left) and an example of compound gear trains (right). Source: http://www.upv.es/vltmodels/index.html

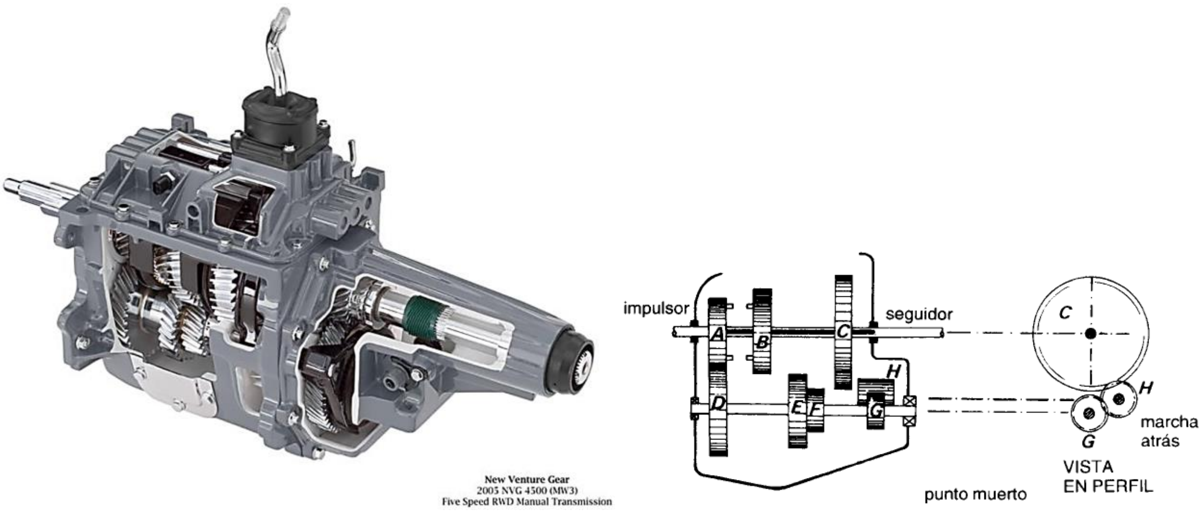


Figure 19. Left: a gearbox as an example which shows how important transmission factor is in mechanisms (the top bar is the shift lever). Right: configuration of neutral gear in a gear box. Source: http://www.upv.es/vltmodels/index.html

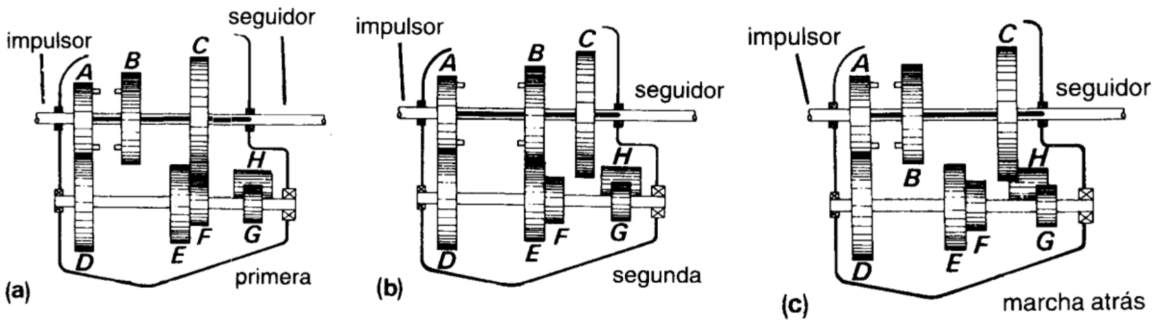


Figure 20. First gear's transmission factor (left) : TF(AC) = TF(AD)*TF(DF)*TF(FC) = TF(AD)*(+1)*TF(FC); 2nd gear (middle): TF(AB) = TF(AD)*TF(DE)*TF(EB) = TF(AD)*(+1)*TF(EB); reverse gear (right): TF(AC) = TF(AD)*TF(DG)*TF(GH)*TF(HC). Then, Sign(TF(1st and 2nd)) = - Sign(TF(Reverse)); TF(1st) < TF(2nd). Source: http://www.upv.es/vltmodels/index.html

Apart from gears, there are some very common mechanisms because of its functionality and its relative ease of being built, used, and maintained. 4-bar and 3-bar can be observed in Doc #1 > Section 5.3.2.

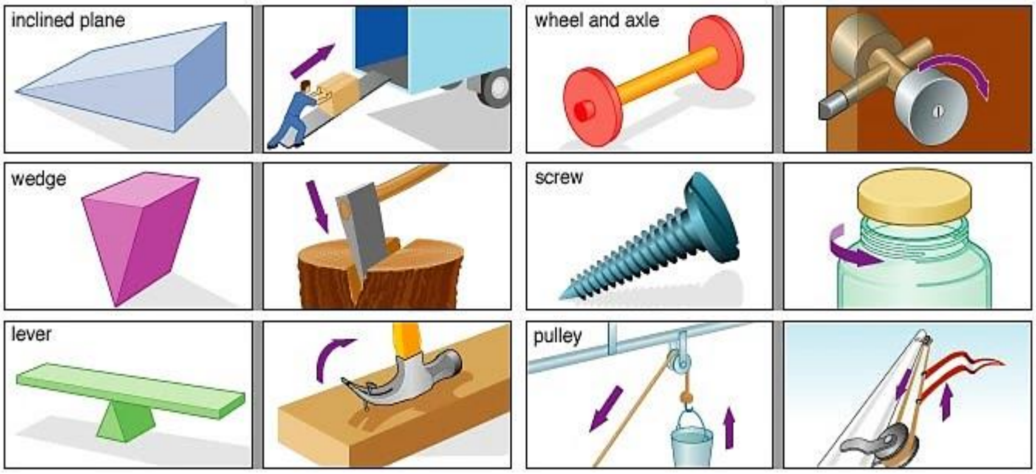


Figure 21. The 6 basic mechanisms. Source: Encyclopedia Britannica® 2006.

Four-bar linkage or **four-bar** or **quadrilateral linkage** is a group of 3 moving pieces where two of them are jointed to the piece fixed to the ground. So, it is defined by 4 pieces and 4 joints. In this mechanism, Grashoff's law says that there is a necessary and enough condition to let at least one of the pieces make a complete revolution with respect to other of the three pieces: s + l <= p + q, where letters are the lengths of the smallest (s), longest (l) and other two pieces (p, q). A crank is a side link which revolves relative to the frame is called a crank. A rocker is any link which does not revolve is called a rocker.

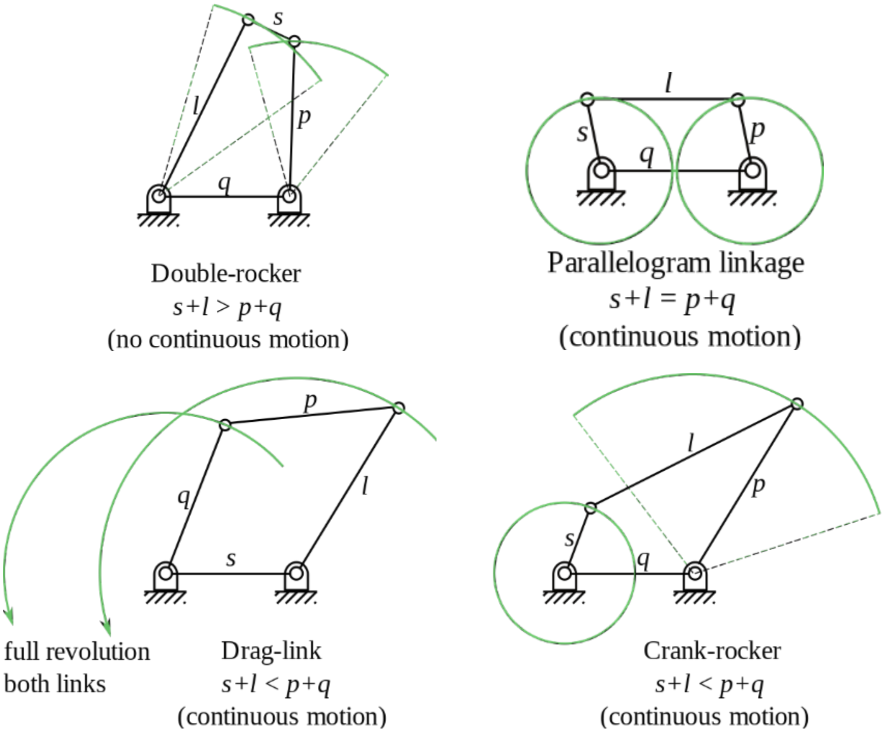


Figure 22. Four-bar linkage depending on link's length. Source: Wikipedia.

Three-bar linkage or **triangle of variable side** is a group of 4 pieces where two of them conforms a side which can vary its length. There are 4 joints. Usually, both conform a shock absorber or a pneumatic cylinder mechanism.

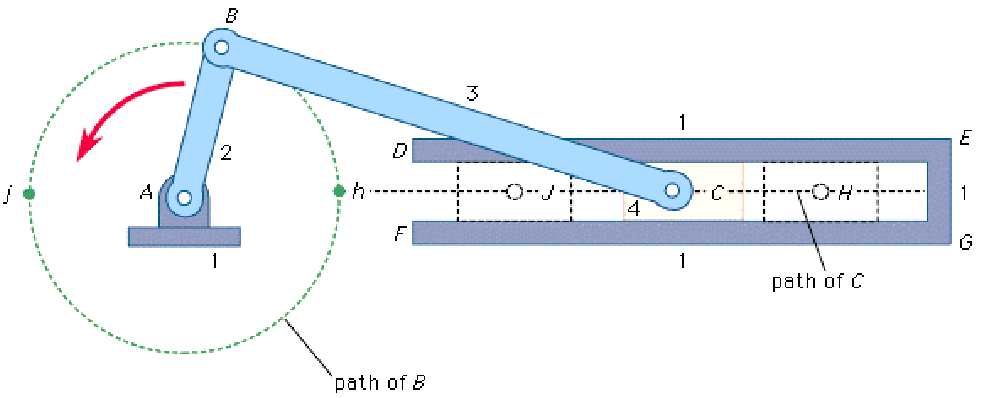


Figure 23. Three-bar linkage. Source: "Machines and mechanisms. Applied kinematic analysis", David H Myszka

Straight-line mechanisms. Straight-line mechanisms cause a point to travel in a straight line without being guided by a flat surface.

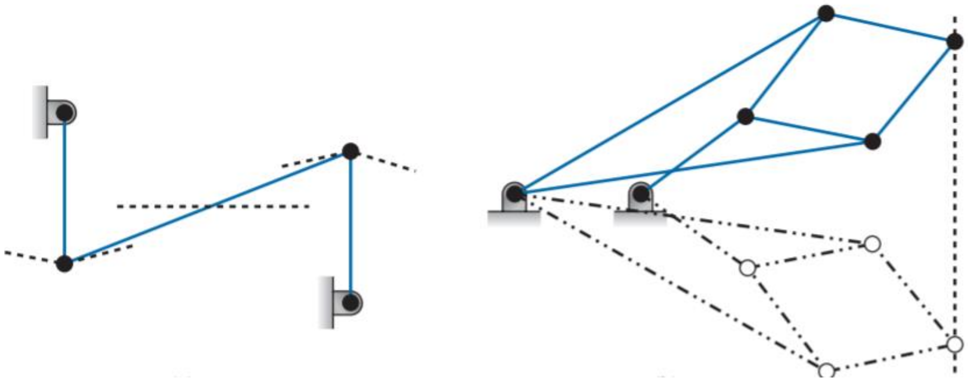


Figure 24. Straight-line mechanisms. Source: "Machines and mechanisms. Applied kinematic analysis", David H Myszka.

Pantograph has 2 degrees of freedom.

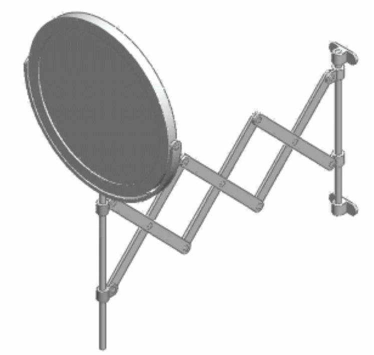


Figure 25. Pantograph. Source: Wikipedia

Parallelogram mechanisms. Mechanisms are often comprised of links that form parallelograms to move an object without altering its pitch. These mechanisms create parallel motion for applications such as balance scales, glider swings, jalousie window, …

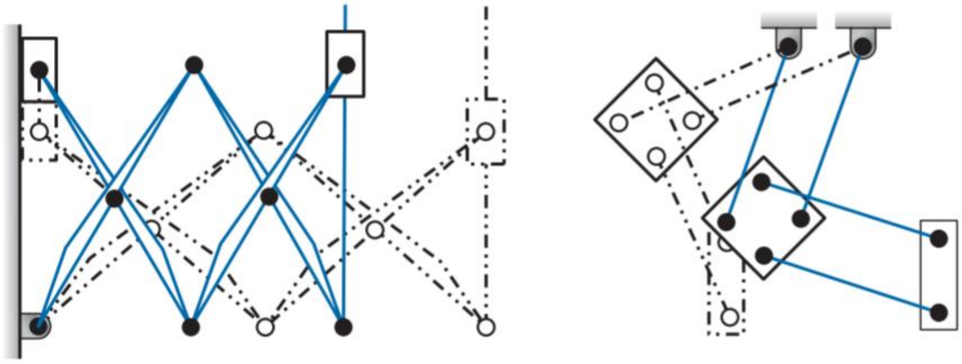


Figure 26. Parallelogram mechanisms. Source: "Machines and mechanisms. Applied kinematic analysis", David H Myszka.

Quick-return mechanisms exhibit a faster stroke in one direction than the other when driven at constant speed with a rotational actuator. They are commonly used on machine tools that require a slow cutting stroke and a fast return stroke**.**

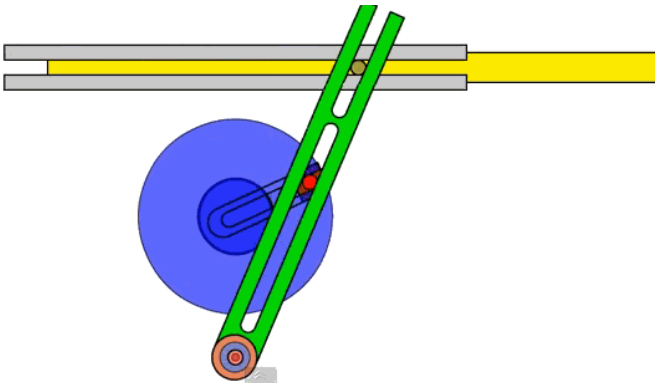


Figure 27. Quick-return mechanism. Source: Wikipedia.

Scotch yoke mechanism. A scotch yoke mechanism is a common mechanism that converts rotational motion to linear sliding motion, or vice versa

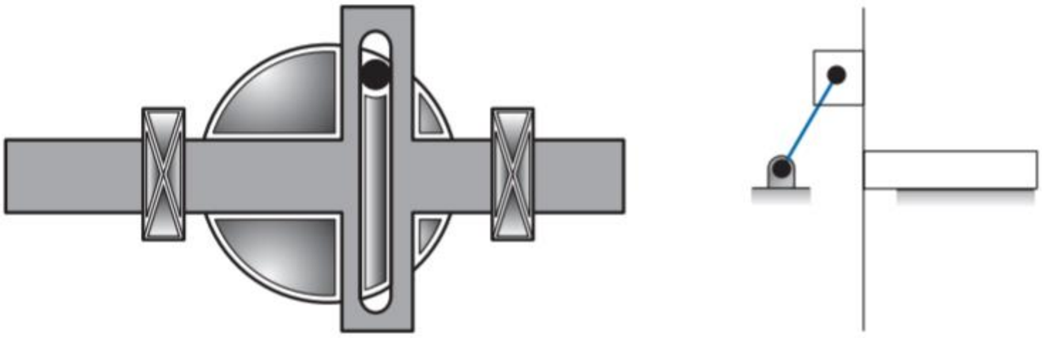
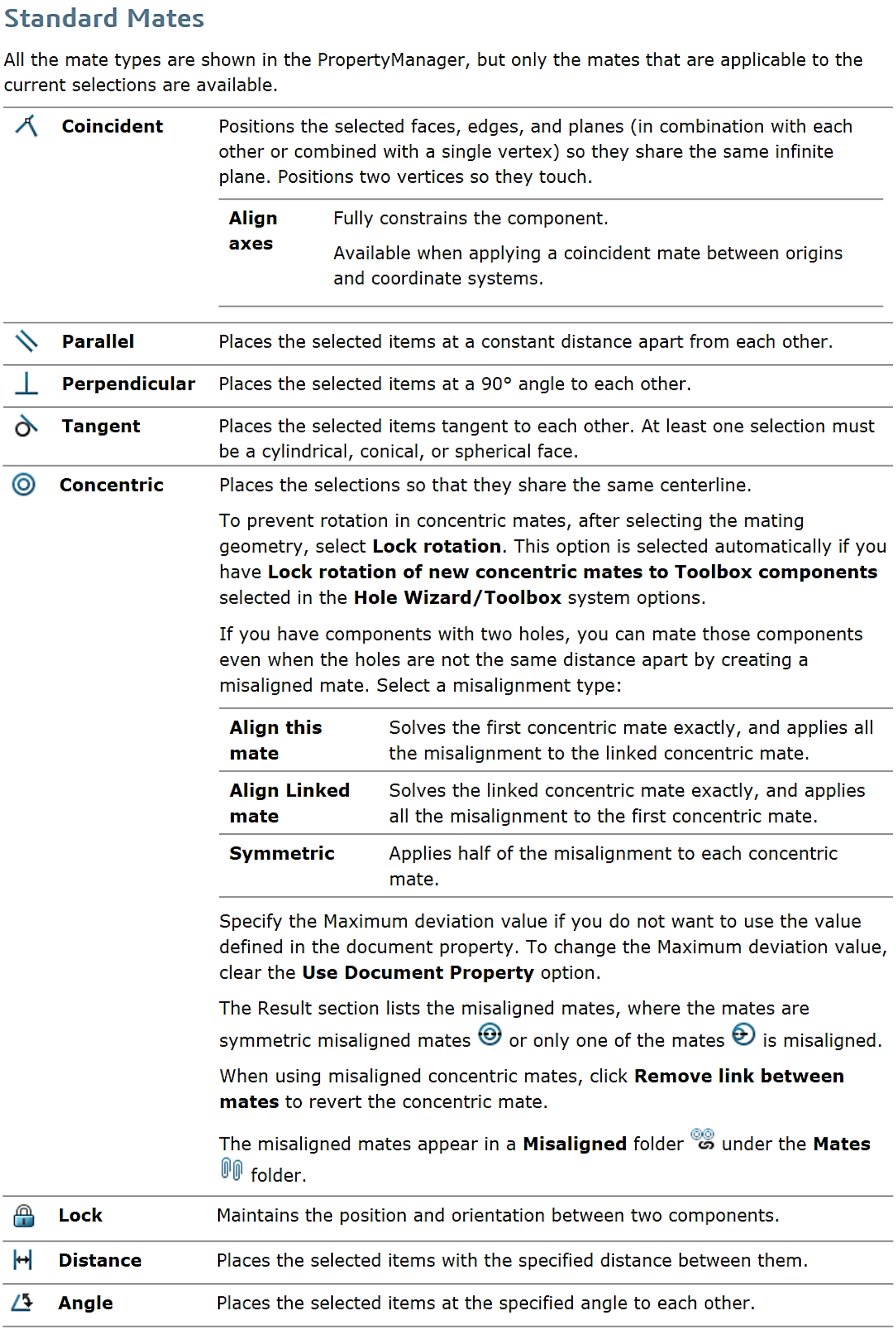


Figure 28. Scotch yoke mechanism. Source: "Machines and mechanisms. Applied kinematic analysis", David H Myszka

3. CAD & CAE USED PROGRAMS IN THE THESIS. MATES AND CONSTRAINTS

3.1. CAD PROGRAM: SOLIDWORKS®®. DESIGN & MATES

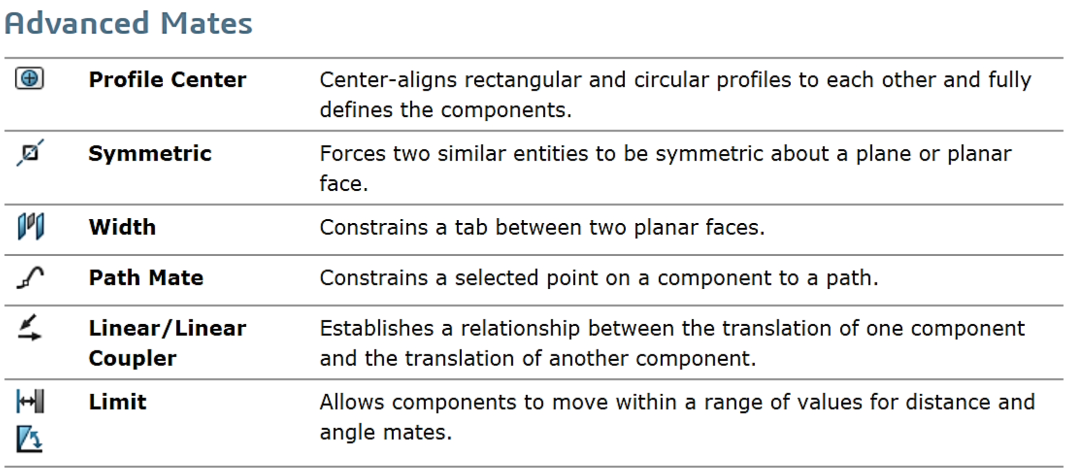


Standard Mates

All the mate types are shown in the PropertyManager, but only the mates that are applicable to the current selections are available.

<table>
<tr><td>Coincident</td><td>Positions the selected faces, edges, and planes (in combination with each other or combined with a single vertex) so they share the same infinite plane. Positions two vertices so they touch.
<table>
<tr><td>Align axes</td><td>Fully constrains the component.
Available when applying a coincident mate between origins and coordinate systems.</td></tr>
</table></td></tr>
<tr><td>Parallel</td><td>Places the selected items at a constant distance apart from each other.</td></tr>
<tr><td>Perpendicular</td><td>Places the selected items at a 90° angle to each other.</td></tr>
<tr><td>Tangent</td><td>Places the selected items tangent to each other. At least one selection must be a cylindrical, conical, or spherical face.</td></tr>
<tr><td>Concentric</td><td>Places the selections so that they share the same centerline.
To prevent rotation in concentric mates, after selecting the mating geometry, select Lock rotation. This option is selected automatically if you have Lock rotation of new concentric mates to Toolbox components selected in the Hole Wizard/Toolbox system options.
If you have components with two holes, you can mate those components even when the holes are not the same distance apart by creating a misaligned mate. Select a misalignment type:
<table>
<tr><td>Align this mate</td><td>Solves the first concentric mate exactly, and applies all the misalignment to the linked concentric mate.</td></tr>
<tr><td>Align Linked mate</td><td>Solves the linked concentric mate exactly, and applies all the misalignment to the first concentric mate.</td></tr>
<tr><td>Symmetric</td><td>Applies half of the misalignment to each concentric mate.</td></tr>
</table>
Specify the Maximum deviation value if you do not want to use the value defined in the document property. To change the Maximum deviation value, clear the Use Document Property option.
The Result section lists the misaligned mates, where the mates are symmetric misaligned mates or only one of the mates is misaligned.
When using misaligned concentric mates, click Remove link between mates to revert the concentric mate.
The misaligned mates appear in a Misaligned folder under the Mates folder.</td></tr>
<tr><td>Lock</td><td>Maintains the position and orientation between two components.</td></tr>
<tr><td>Distance</td><td>Places the selected items with the specified distance between them.</td></tr>
<tr><td>Angle</td><td>Places the selected items at the specified angle to each other.</td></tr>
</table>

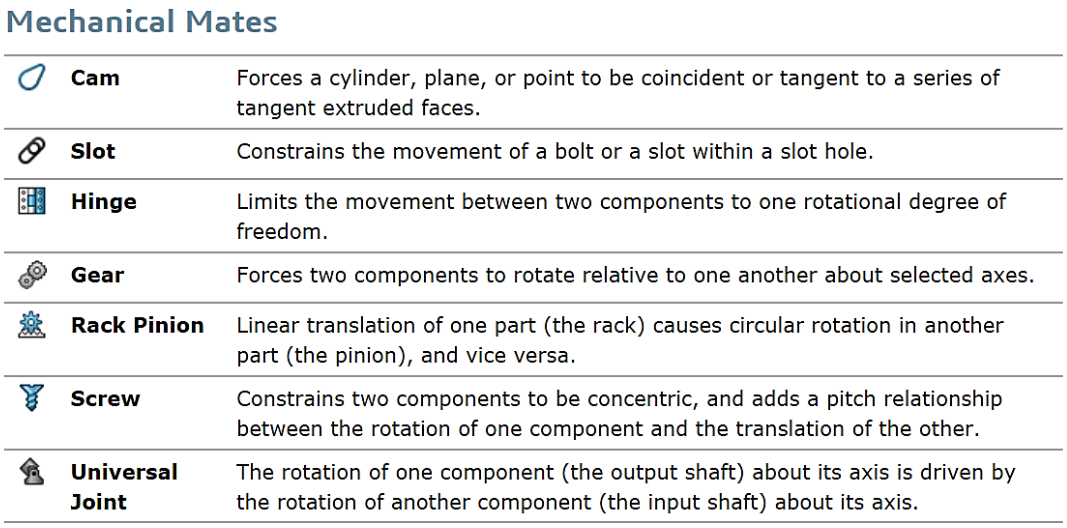
Figure 29. All possible standard mates in SolidWorks® 2020. Source: SolidWorks® Online Help Center



Advanced Mates

Profile Center	Center-aligns rectangular and circular profiles to each other and fully defines the components.
Symmetric	Forces two similar entities to be symmetric about a plane or planar face.
Width	Constrains a tab between two planar faces.
Path Mate	Constrains a selected point on a component to a path.
Linear/Linear Coupler	Establishes a relationship between the translation of one component and the translation of another component.
Limit	Allows components to move within a range of values for distance and angle mates.

Figure 30. All possible advanced mates in SolidWorks® 2020. Source: SolidWorks® online help center



Mechanical Mates

Cam	Forces a cylinder, plane, or point to be coincident or tangent to a series of tangent extruded faces.
Slot	Constrains the movement of a bolt or a slot within a slot hole.
Hinge	Limits the movement between two components to one rotational degree of freedom.
Gear	Forces two components to rotate relative to one another about selected axes.
Rack Pinion	Linear translation of one part (the rack) causes circular rotation in another part (the pinion), and vice versa.
Screw	Constrains two components to be concentric, and adds a pitch relationship between the rotation of one component and the translation of the other.
Universal Joint	The rotation of one component (the output shaft) about its axis is driven by the rotation of another component (the input shaft) about its axis.

Figure 31. All possible mechanical mates in SolidWorks® 2020. Source: SolidWorks® Online Help Center.

When selecting a mate rack pinion, it is needed to select a circular face of the pinion and to select a straight line of the rack. In both cases, axis of rotation is being selected.

3.2. CAE PROGRAMS. CONSTRAINTS & SIMULATION

3.2.1. SOLIDWORKS® 2007 SP5.0 > COSMOSMOTION 2007 ADD-IN

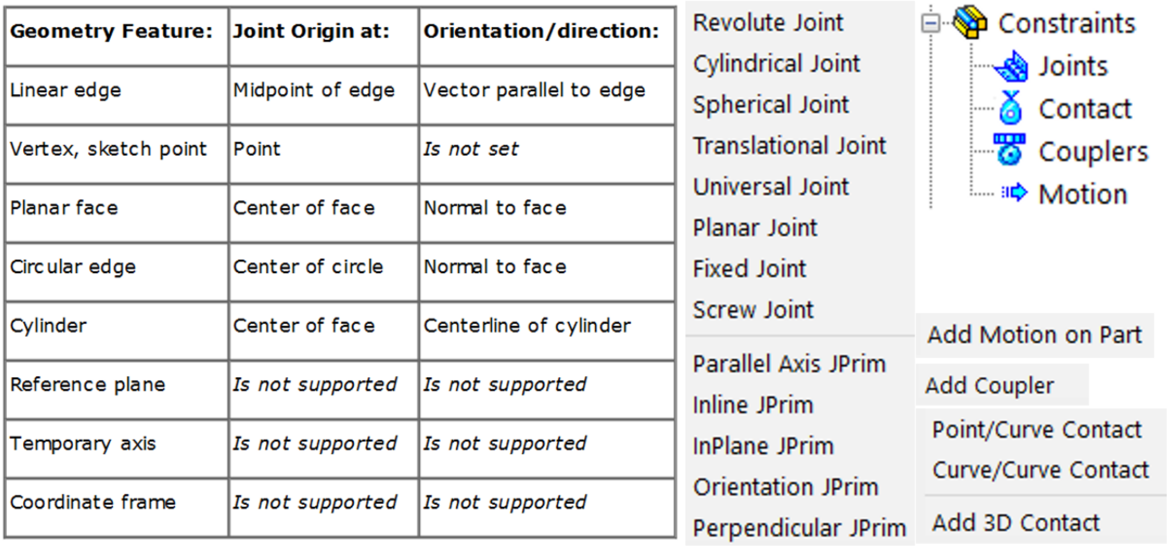
SolidWorks®2007, a CAD program, has several add-ins. One of them is COSMOSMotion 2007, a CAE program. In next versions of SolidWorks® there is no more this add-in. By using that add-in, creating constraint to do the kinematic analysis is very easy and fast. However, it is not automatic: knowledge about how to create self-aligning is needed. The use of SolidWorks® 2007 and COSMOSMotion is very easy to understand because of its simpleness and icons. The more mechanisms are created, the better the command of the program will be. Even, if needed, in 'Help' tab can be found more information.

There is an important aspect to consider. It was created to be run in Windows Vista. If it is run in Windows Vista, 7 or 8, error messages are not supposed to pop up. At least in Windows 10, every time a command like "save as" or "open" is clicked, two or three errors appear. They say "an unsupported operation was attempted". There are ways of reducing the nº and probability of an error message to pop up. In addition to this, there is another important problem. When running this program, graphic card seems not being used because of the year the program was created. So, it is highly recommended (1) to visualize piece with "Shade" (not "Shade with Wire"), as well as (2) to save very often because it might stop running properly. It can close by itself or get blocked. If this happens, it is needed to click "End Task" in "Task Manager" Windows app. Other problem is that very large assemblies are difficult to be edited, because of response time of display.

When using COSMOSMotion to create constraints, it is recommended to change color and size of icons. It can be found in the left organizational tree of Cosmos > Right click on "Motion Model" tab > "Display parameters" > "Entity part", and whenever the color of the part is selected, click on "apply to existing entities" > Click on "Apply". But, once constraints are created, they might not change of color. It is needed to change colors before creating the constraints to create

If something is not very accurate, the traffic signal icon can be clicked. It will refresh the display and the results

To properly define constraints, it is recommended to select next geometry features. For instance, if an arc (circular edge) is not selected but the cylinder, the origin and direction of the created joint may not be as wanted. To select the direction and origin of the joint, chose a parallel edge or a perpendicular plane to the axis of the joint. Thus, origin will be in the middle of the feature selected.



Geometry Feature:	Joint Origin at:	Orientation/direction:
Linear edge	Midpoint of edge	Vector parallel to edge
Vertex, sketch point	Point	*Is not set*
Planar face	Center of face	Normal to face
Circular edge	Center of circle	Normal to face
Cylinder	Center of face	Centerline of cylinder
Reference plane	*Is not supported*	*Is not supported*
Temporary axis	*Is not supported*	*Is not supported*
Coordinate frame	*Is not supported*	*Is not supported*

Figure 32. Predefined features to define joints and all possible constraints in COSMOSMotion 2007. Source: COSMOSMotion 2007 Help Center.

3.2.2. RECURDYN V8R5

Recurdyn is a CAE program. This professional program is intuitive and fast to use. A lot of tools and procedures are well implemented. It very fast with respect to other programs. Next, there is a description written by RecurDyn's developer business company to introduce its product.

"RecurDyn is Integrated Multi-Discipline (IMD) Computer Aided Engineering (CAE) software. Its primary purpose is Multi-Flexible Body Dynamics (MFBD), which is the prediction of the motion of mechanisms and devices composed of solid bodies, in which the bodies can be modeled as either rigid or flexible bodies. RecurDyn contains many components, such as Finite Element Analysis (FEA), Optimization Design, Automatic Control, Fatigue Analysis, as well as rigid Multibody Dynamics.

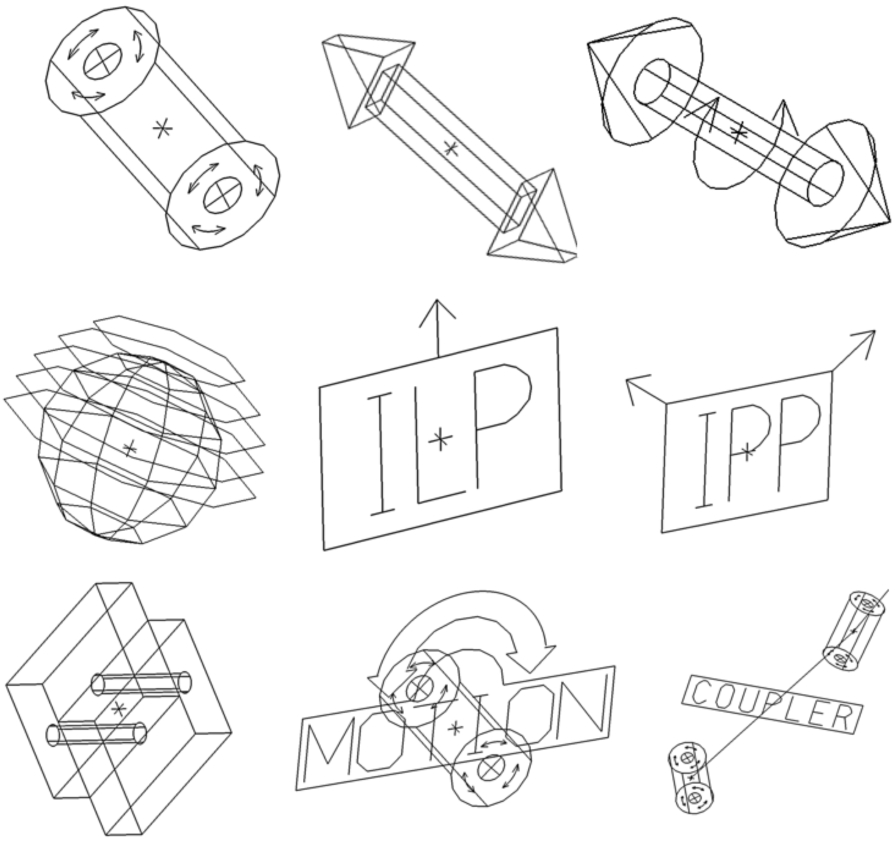
RecurDyn has superior calculation efficiency because it is based on a recursive formulation. Therefore, RecurDyn gives the best performance in large-scale multibody problems (systems with many bodies), including mechanical systems in which complex contact between bodies is very significant. RecurDyn also features a natural, Windows-based User Interface which is intuitive and easy to use, as well as a custom application development environment which enables users to automate complicated and/or tedious tasks. RecurDyn contains many class-leading technologies:

- Fastest Linear Solver. RecurDyn uses the best-available multi-frontal linear solver. This linear solver is excellent for MFBD models that contain rigid and flexible bodies.
- Fast Contact Between Solid Bodies. Contact is easy to define and simulates rapidly. Contacts can quickly be defined even for imported CAD geometry. RecurDyn's calculation speed for models with contact is exceptionally fast due to FunctionBay's unique contact algorithms that combine efficient and accurate methods from mechanical engineering and computer science.

Intuitive GUI. RecurDyn has an intuitive GUI which is both easy to use for new users and powerful for experienced users. RecurDyn can also easily accept modeling data from CAD software"

Next, all RecurDyn's current available constraints are shown, as well as forces. Notice there are many more constraints than exposed in COSMOSMotion. In addition, notice that RecurDyn considers 1 screw joint removes 2 DOF, not 1 as COSMOSMotion does.

Figure 33. All types of joints, contacts, and forces available in RecurDyn.



RecurDyn

COSMOSMotion 2007

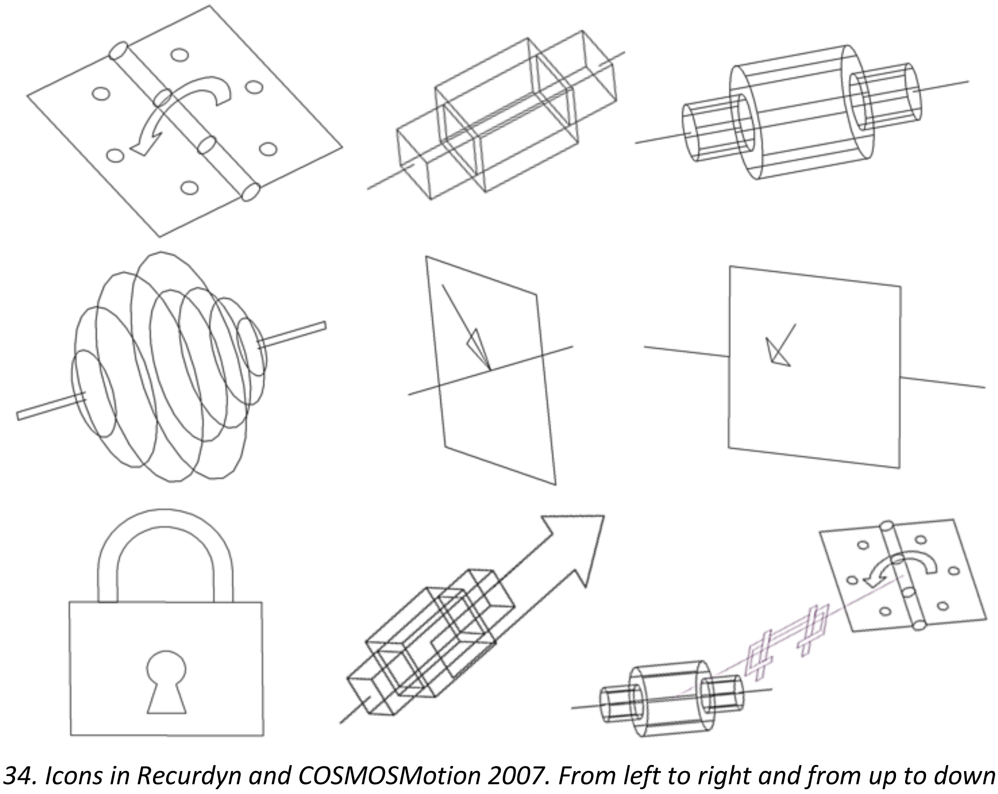


Figure 34. Icons in Recurdyn and COSMOSMotion 2007. From left to right and from up to down: R, P, C, Sph, IL, IP, F, "Motion in the properties of a R, P or C joint", and "Coupler between R, P or C joints".

4. MECHANISMS' MOBILITY. CONSTRAINTS

4.1. TYPES OF CONSTRAINTS

Each constraint is associated with a unique set of geometric entities, including points and axes. Some of the needed features to define a joint is an origin point and a plane to which it will be perpendicular. It restricts the relative movement between each two bodies. So, it defines the motion a piece can have with the other one. Each constraint restricts bodies' trajectories in a specific way. For instance, if a revolute joint is applied to a free body, the degrees of freedom of that body are reduced from six to one. This DOF only lets the body to rotate around its axis with respect the other body.

Not only understanding the meaning of DOF is critical to select the appropriate constraints to define a mechanism's ability to move. It is also needed to visualize DOF from a vectorial point of view (rotation and translation). Depending on the type and direction of each joint in a mechanism affect to M_{ap} and NRC.

By default, usually constraints are idealized: they are infinitely rigid, do not have mass, and do not have any clearances or slop. Global settings can be used to make the basic joints flexible instead of rigid. This is called a **bushing**. When defining a bushing, it is needed to determinate rotational and translational coefficient of rigidity in the three axes. The more rigidity coefficient, the less degree of freedom.

When self-aligning a mechanism in this Thesis, only joints which appear in Table of Reshetov will be accepted.

4.1.1. JOINTS

A joint reduces M_{calc} by the nº of DOF it constrains/removes, by $6 – f\_i$.

When defining a joint, two groups can be differenced:

- **General joints** are used to constrain the relative motion of a pair of rigid bodies by physically connecting them.
- **Primitive joints** are used to enforce standard geometric constraints. Primitive joints place a restriction on relative motion, such as restricting one part to always move parallel to another part. The primitive joints, except in line and in plane, do not have physical counterparts as the general joints do. However, the user combines primitive joints to define a complex constraint that cannot be modeled using the general joints. In fact, by using the primitive joints, any of the general joints can be created.

Next, there is a brief description of some types of joints.

In Plane joint**, IP.** A point is coincident to a plane.

In Line joint**, IL.** A line is coincident to a plane.

Spherical joint**, Sph**. The spherical joint allows free rotation about a common point of one rigid body with respect to another rigid body. The origin location of the spherical joint determines the point about which the joint's rigid bodies can pivot freely with respect to each other.

Planar joint**, PL**. The planar joint allows a plane on one rigid body to slide and rotate in the plane of another rigid body. It is not used because it does not let three-dimensional movements.

> The origin location of the planar joint determines a point in space through which the joint's plane of motion passes. The rotational axis of the planar joint is parallel to the orientation vector.

Cylindrical joint**, C**. The cylindrical joint allows both relative rotation and relative translation of one rigid body with respect to another rigid body.

> The origin of the cylindrical joint can be located anywhere along the axis about which the rigid bodies can rotate or slide with respect to each other. Orientation of the cylindrical joint defines the direction of the axis about which the rigid bodies can rotate or slide along with respect to each other. The rotational/translational axis of the cylindrical joint is parallel to the orientation vector and passes through the origin.

Universal joint or **Hooke** joint or **Cardan** joint**, U**. The universal joint allows the rotation of one rigid body to be transferred to the rotation of another rigid body. This joint is particularly useful to transfer rotational motion around corners, or to transfer rotational motion between two connected shafts that are permitted to bend at the connection point (such as the drive shaft on an automobile).

> The origin location of the universal joint represents the connection point of the two rigid bodies. The two shaft axles identify the center lines of the two rigid bodies connected by the universal joint. Note that COSMOSMotion uses rotational axles parallel to the identified rotational axles but passing through the origin of the universal joint

Next, it can be observed how to self-align three bodies which conforms a universal joint. It is need two C and 2 IP. One C has to be opposite one IP, and this is done for both axes.

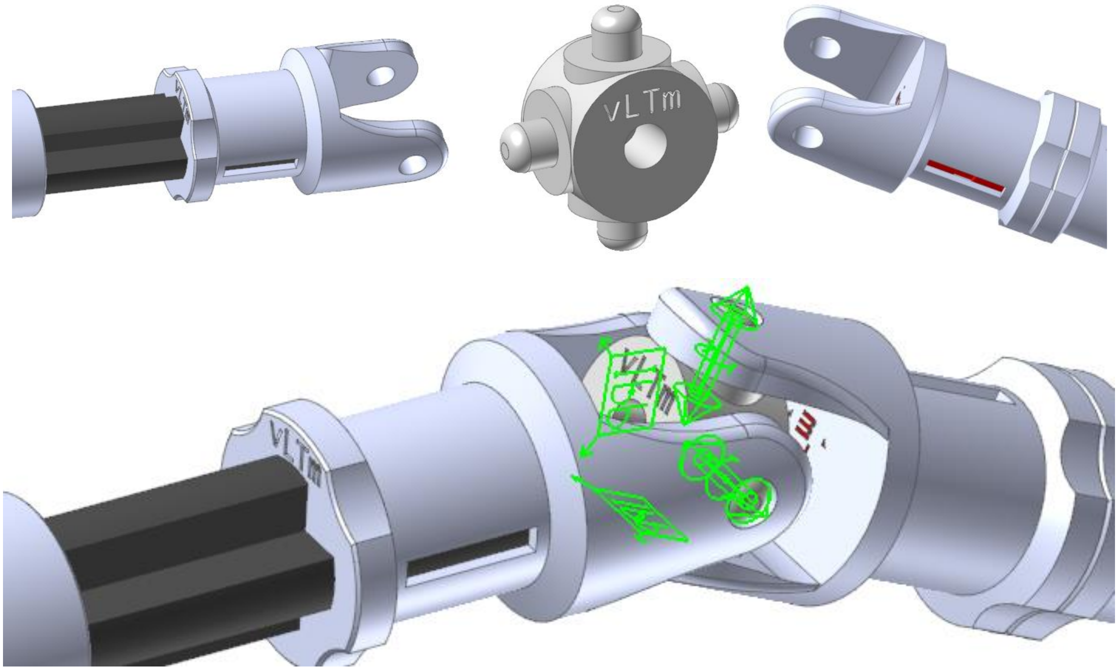


Figure 35. Constraints to simulate a universal joint.

Prismatic joint**, P**: The translational joint allows one rigid body to translate along a vector (a line) with respect to a second rigid body. The rigid bodies may only translate, not rotate, with respect to each other.

> The location of the origin of a translational joint with respect to its rigid bodies does not affect the motion of the joint but does affect the reaction loads of the joint. The orientation of the translational joint determines the direction of the axis along which the rigid bodies can slide with respect to each other (axis of translation).

Revolute joint**, R.** The revolute joint allows the rotation of one rigid body with respect to another rigid body about a common axis.

> The revolute joint origin can be located anywhere along the axis about which the joint's rigid bodies can rotate with respect to each other. The rotational axis of the revolute joint is parallel to the orientation vector and passes through the origin

Helicoidal or **screw** joint**, H.** The screw joint removes one degree of freedom. It constrains one rigid body to rotate as it translates with respect to another rigid body. Very often, the screw joint is used with a cylindrical joint. The cylindrical joint removes two translational and two rotational degrees of freedom. The screw joint removes one more degree of freedom by constraining the translational motion to be proportional to the rotational motion. the pitch is the amount of translational displacement of the two rigid bodies for each full rotation of the first rigid body. The displacement of the first rigid body relative to the second rigid body is a function of the rotation of the first rigid body about the axis of rotation. That is why the sum of 0.5 rotational DOF and 0.5 translational DOF gives screw joint 1 DOF: rotation and translation movement are needed at the same time to produce motion.

Fixed, F. The fixed joint locks two rigid pieces together so they cannot move with respect to each other. Removes all 6 DOF and rigidly connects parts. Parts connected this way are treated as a single part with mass properties equal to the total of the two parts' mass properties. A real-world example of a fixed joint is a weld or glue that holds two parts together. However, it might be needed to consider one fixed joint if both pieces are redundant in the model.

> The location and orientation of its origin does not affect the outcome of the simulation.

4.1.1.1. TABLE OF RESHETOV

In the next two pages, the **Table of Reshetov** is introduced. Looking at this table, it is possible to go from a joint designed in the paper to something which can be implemented in the "physical world". In consequence, when self-aligning a mechanism in this Thesis, only joints which appear in Table of Reshetov will be accepted. The other constraints (coupler, contact and motion) are not in this table, and obviously they take place in the "physical world" so they also will be used in this Thesis.

In table 1.1, when a bearing is eroded, it is converted into one bearing located to the right of the thick black line. But it will not be considered while constraining the mechanism.

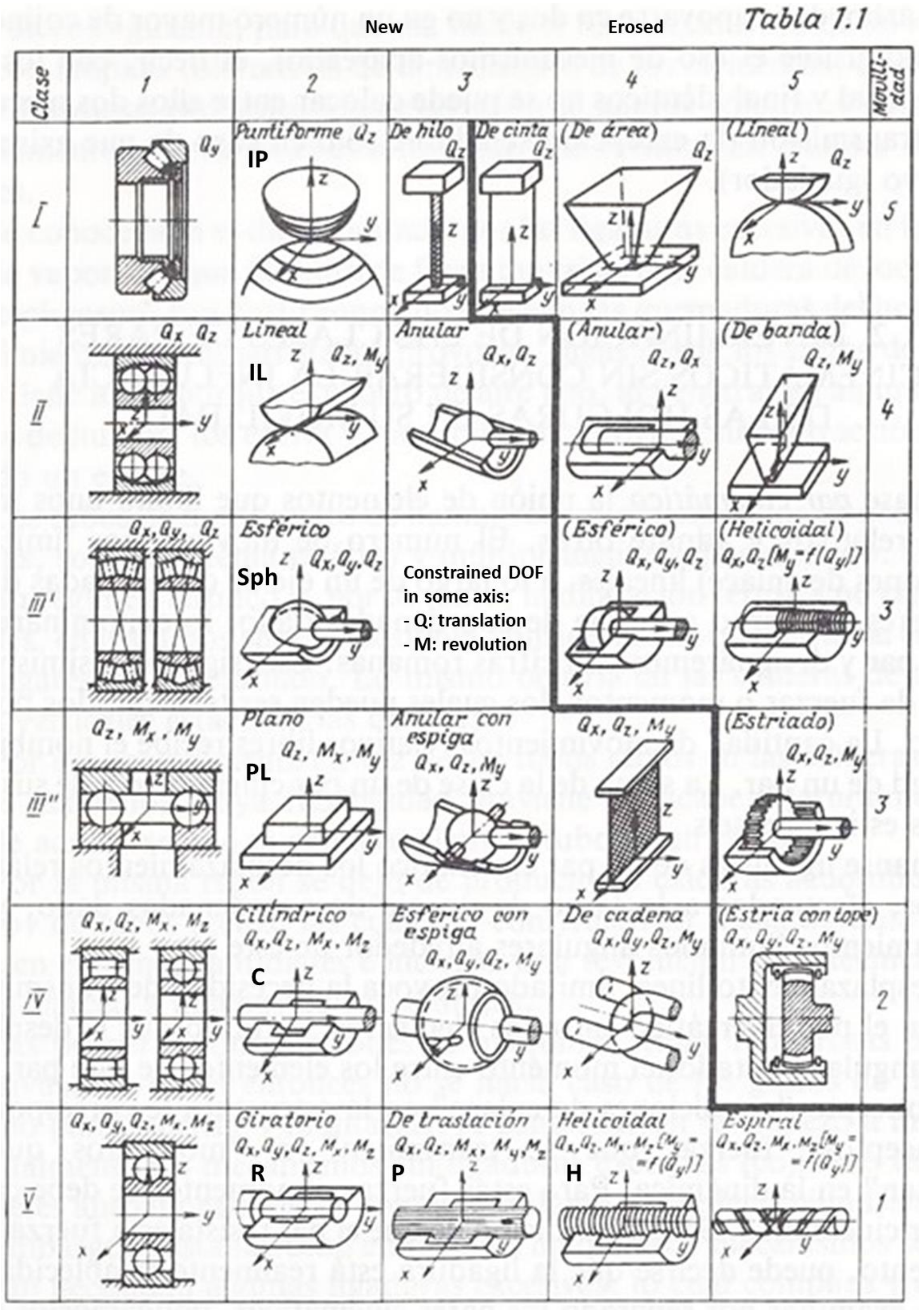


Figure 36. Table 1.1 of Reshetov. Some joints can be bearings: R, P, H, C, Sph, IL, IP. Source: Reshetov, L. (1982). In Mir Publishers (Ed.), "Self-aligning mechanisms".

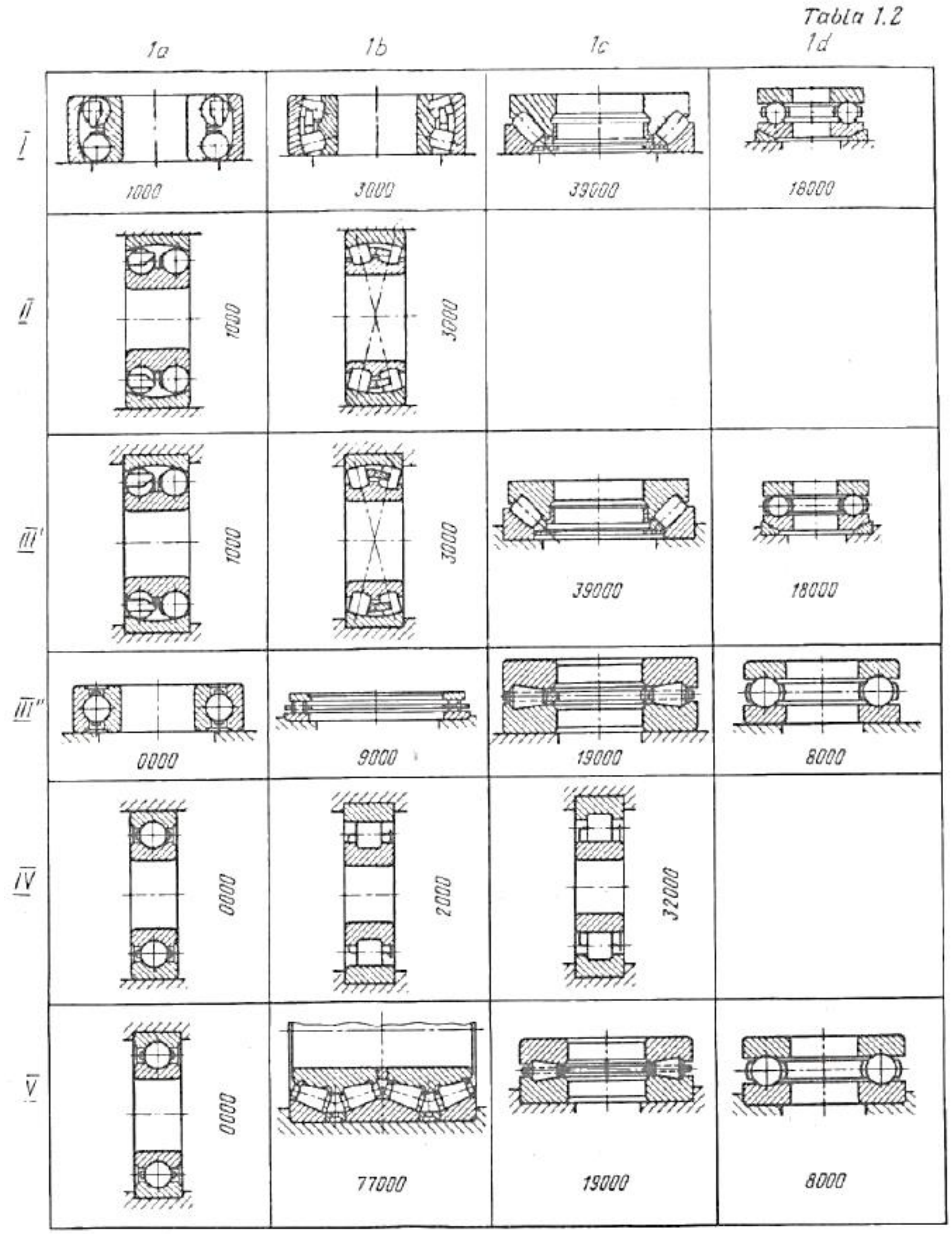


Figure 37. Table 1.2 of Reshetov. A detail of bearings. Source: Reshetov, L. (1982). In Mir Publishers (Ed.), "Self-aligning mechanisms".

4.1.1. COUPLERS

Couplers are used to provide proportional movement between translational and/or rotational movements on revolute, cylindrical, and translational joints. It only can be stablished between these three types of joints. Although a couple constraint can only couple two joints, more than one coupler can use the same joint to be defined. The way to understand how a coupler is defined is as follows: when joint ... rotates/translates ... (º or mm), other joint ... rotates/translates ... (º or mm). That is a ratio. Couplers are useful to simulate pulleys, belt, chains, sprockets, gears, … For example, if a revolute joint is being coupled with a translational joint, 90 degrees could be specified for the revolute joint, and 100 mm for the translational joint. This means that for every 90 degrees the revolute joint rotates, the translational joint moves 100 mm, or 1 degree of rotation results in 90/100 = 0.9 mm of translation.

A coupler does not differentiate between a driving joint and a driven joint. A joint in the couple may at one point in time be driving the other joint. A potential for locking does exist if both joints in a coupler are driven by motion generators. The coupler may conflict with the motion generator.

One coupler removes 1 additional DOF from the motion model, so M_{calc} and M_{ap} are reduced by 1, because make 1 DOF dependent of the other.

In next figure, obviously, only one coupler constraint must be user-defined between the two revolute joints. Thanks to that, when piece nº 2 rotates, piece nº 3 will do too in a proportional way. Usually, total nº of coupler constraints for gear, chain, belts … is the nº of wheels or chain/belt lines minus 1.

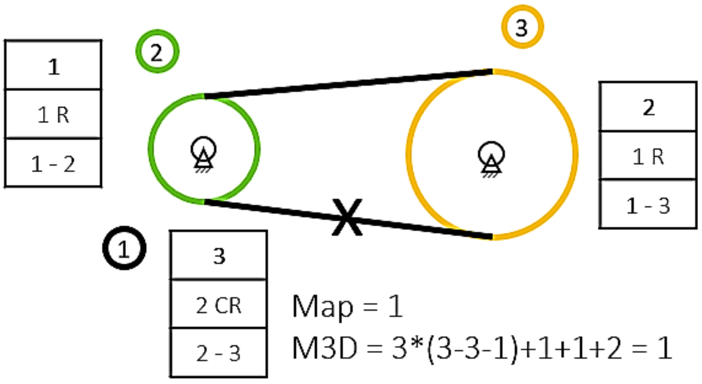


Figure 38. A belt is defined by a coupler constraint.

4.1.2. CONTACTS

Contact or **cam** constraints are used to simulate contact between a point and a curve or a surface, or between two curves or two surfaces.

- **Point - curve**. Restricts a point on one rigid body to lie on a curve on a second rigid body.
- **Curve - curve**. Constrains one curve to remain in contact with a second curve.
- **Intermittent curve – curve.** Applies a force to prevent curves from penetrating each other. Only active if the parts are touching

- **3D Contact**. Applies a force to prevent bodies from penetrating each other. Only active if the parts are touching

Only 'point - curve' and 'curve - curve' contacts impose DOF restrictions on the connected parts and are true constraints**.** Intermittent curve-curve and 3D contacts are included in this group, but do not impose fixed degrees of freedom between parts (only forces between the bodies). If one 'point - curve' or one 'curve - curve' contacts (only these contact constraints) is added to the model, M_{calc} is reduced by 2 whilst M_{ap} does not change.

4.1.3. MOTION

Motion. It can be created as a force or a property in the properties of the joint. By only using this second option, a better control of motion and of calculated mobility can be achieved. It enforces displacements, velocities, and acceleration, so it removes degrees of freedom. When adding 1 'motion' in the properties of the joint, M_{calc} and M_{ap} are reduced by 1. It is user-defined by writing mathematical equations: displacement, velocity, acceleration, functions, … in some rotation or translation axis values can be used.

The purpose of defining 'motion' constraints is to, once the mechanism is self-aligned, going from a M_{calc} and $M_{ap} > 0$ to =0. Of course, it is not needed to get = 0. The simulation has to represent the real behavior of the mechanism. When motion is defined in joint's properties, the moving piece becomes a piece with desmodromic movement.

These equations can be constants, step, harmonic, spline … functions in general. Step function is very useful. All values of motion are with respect to the reference system of the joint at the initial position. When writing the function, a "d" is needed to indicate "degrees". To run a simulation is needed to know if it is a + or -sign (it just changes direction of motion).

- **Constant**: displacement, angular displacement, velocity, angular velocity, acceleration, angular acceleration, initial displacement, initial velocity.
- **Step:** STEP (x, x0, h0, x1, h1)
  - x. The independent variable. It can be a function expression.
  - x0. A real variable that specifies the x value at which the STEP function begins.
  - x1. A real variable that specifies the x value at which the STEP function ends.
  - h0. The initial value of the step.
  - h1. The final value of the step.

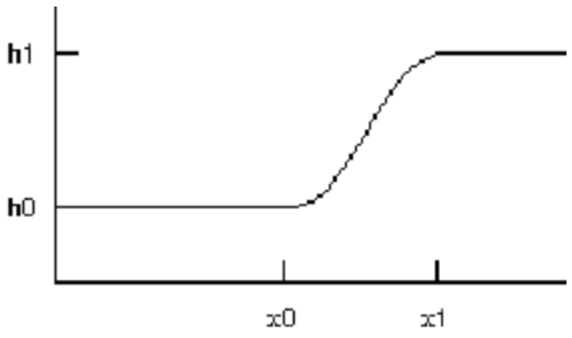


Figure 39. A step function to define motion in a joint. Source: COSMOSMotion 2007 Help Center.

For example, a step function inside another one: first run motion function is the one more 'inside'. For example, STEP(TIME,1,STEP(TIME,0,0D,1,+120D),2,-120D): first, from second 0 to 1, it moves from 0 to 120 degrees with respects the initial position of the system reference of the constraint; then, from second 1 to 2, it turns up to 240 degrees to the opposite location. But if STEP(TIME,1,STEP(TIME,0,0D,1,+120D),2,0), it moves 120 degrees and after it comes back to its initial position.

- **Harmonic:** f(t) = A*sin(w*(t-T0) - j) + B, where
  - A = Amplitude of the value
  - w = Frequency
  - T0 = Offset Time
  - j = Phase Shift
  - B = Average value
  - Time Offset. Specify the offset from zero where the function starts at its average value.
  - Phase Shift. Specify the phase shift for the function.
  - Average. Specify the average value of the function. The function oscillates about this value.

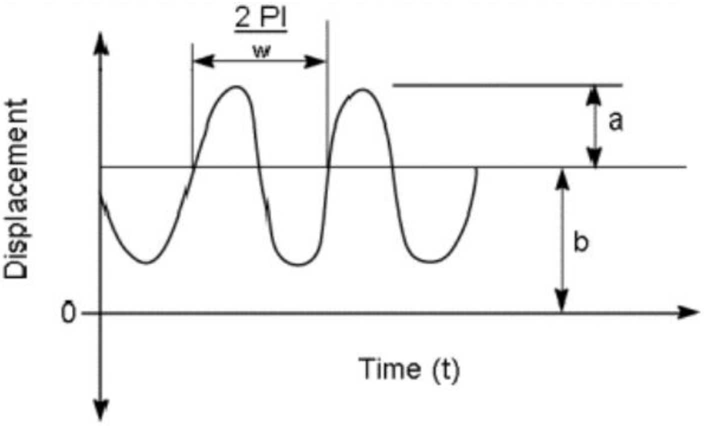


Figure 40. A harmonic function to define motion in a joint. Source: COSMOSMotion 2007 Help Center.

- **Spline**

The spline function lets define a motion generator by specifying discrete displacement values at specific points in time. During the simulation, COSMOSMotion interpolates the discrete points using one of two types of splines, which results in a smooth, continuous curve.

  - X Units. Enter time values separated by commas. For example: 0.0, 0.1, 0.2, 0.3, 0.4, 0.5.
  - Y Units. Enter data values separated by commas. The data values must correspond to the time values, and the same number of values must be present.
  - Data ID. If a spline function has already been defined, the same function can be applied to this motion by selecting the ID. The new selection creates a new spline.
  - Type of spline interpolation. CUBSPL and AKIPL.

  - Load from file. To select to load data point values from a file. The file should contain one data point per line. The data point consists of two values: the time and the value at that time. It is needed to use commas or spaces as separators between the values.

It is possible to create more than one set of data points and reference them between motions and forces. To use the Data ID list to control the set of points that is being applied. When a curve is created for the first time, the data ID list is set to NEW; when more than one set is created, or when existing sets are modified, it is possible to select from existing sets of data points or to create a new one.

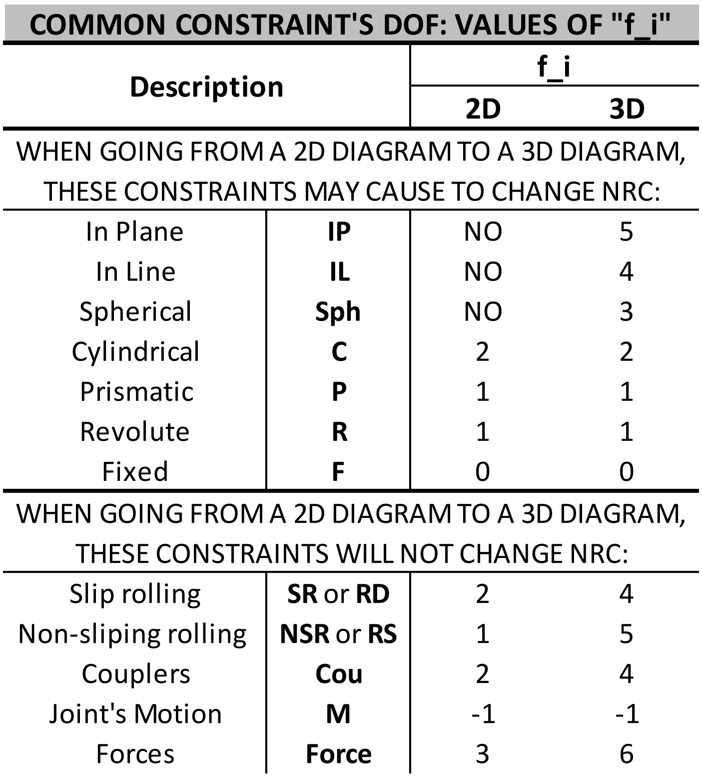
4.2. ABSTRACT TABLE OF THE TYPES OF CONSTRAINTS

Next, there is a summary with all COSMOSMotion's predefined sets of constraints and more. It will be explained through this section. COSMOSMotion Is one of the two CAE program used in this Thesis.

CONSTRAINT'S DOF: VALUES OF "f_i"

When λ=6			**When λ=6**			**Identification**			**f_i**		**Description**
λ - f_i	**Rot**	**Tra**	**f_i**	**Rot'**	**Tra'**	**Type**	**Letter**	**Names**	**2D (λ=3)**	**3D (λ=6)**	
WHEN GOING FROM A 2D DIAGRAM TO A 3D DIAGRAM, THESE CONSTRAINTS MAY CAUSE TO CHANGE NRC:											
JOINTS						**M_calc is reduced by "λ - f_i", & M_ap does not change, so NRC > 0 increases by this nº.**					
1	0	1	**5**	3	2	Primitive	**IP**	In Plane	NO	5	1 point is in a plane.
	1	0		2	3		-				
2	0	2	**4**	3	1	Primitive	**IL**	In Line	NO	4	A line is coincident to a plane.
	2	0		1	3		-				
	1	1		2	2		-				
3	0	3	**3**	3	0	General	**Sph**	Spherical , ball	NO	3	Requires 1 point of each body.
	3	0		0	3		-				
	1	2		2	1		-				
	2	1		1	2	General	PL	Planar	3	NO	Not available for 3D.
4	1	3	**2**	2	0	General	U	Universal			
	3	1		0	2		-				
	2	2		1	1	General	**C**	Cylindrical	2	2	1 axis of Rot and/or 1 Tra.
5	2	3	**1**	1	0	General	**R**	Revolute , pin, hinge	1	1	1 axis of Rot on each part.
	3	2		0	1	General	**P**	Prismatic , translational, slider	1	1	1 axis of Tra on each part.
	2,5	2,5		0,5	0,5	General	**H**	Helicoidal , screw			
6	3	3	**0**	0	0	General	**F**	Fixed , rigid	0	0	No relative movement
WHEN GOING FROM A 2D DIAGRAM TO A 3D DIAGRAM, THESE CONSTRAINTS WILL NOT CHANGE NRC:											
CONTACTS						**M_calc is reduced by "λ - f_i", & M_ap does not change, so NRC > 0 increases by this nº.**					
2	0	2	**4**	3	1		-	Point - Curve			
	0	2		3	1		-	Curve - Curve			
2	1	1	**4**	2	2		**SR** or **RD**	**Slip rolling**	2	4	*(rodadura con deslizamiento)*
1	0,5	0,5	**5**	2,5	2,5		**NSR** or **RS**	**Non-sliping rolling**	1	5	*(rodadura sin deslizamiento)*
0	0	0	**6**	3	3			3D contact	6	6	
COUPLERS (R, P, C)						**NRC doesn't change because M_ap & M_calc are reduced by "λ - f_i".**					
2	1+1	0	**4**	1	3		**Cou**	**Coupler**	2	4	Any coupler (G, CB, …)
							G or **E**	**Gear** *(Engranaje)*	2	4	There is no sliping.
							CB or **CR**	**Chain** / **Belt** *(Correa/Cadena)*	2	4	
MOTION						**NRC doesn't change because M_ap & M_calc are reduced by "λ - f_i".**					
+1	0	1	**-1**	-1	0		**M**	Motion in joint's properties	-1	-1	Motion of translation
	1	0		0	-1				-1	-1	Motion of rotation
FORCES						**M_calc, M_ap and NRC do not change.**					

Figure 41. Table of the value of "f_i" (the nº of degrees of freedom) of some constraints.



COMMON CONSTRAINT'S DOF: VALUES OF "f_i"

Description		**f_i**	
		2D	**3D**
WHEN GOING FROM A 2D DIAGRAM TO A 3D DIAGRAM, THESE CONSTRAINTS MAY CAUSE TO CHANGE NRC:			
In Plane	**IP**	NO	5
In Line	**IL**	NO	4
Spherical	**Sph**	NO	3
Cylindrical	**C**	2	2
Prismatic	**P**	1	1
Revolute	**R**	1	1
Fixed	**F**	0	0
WHEN GOING FROM A 2D DIAGRAM TO A 3D DIAGRAM, THESE CONSTRAINTS WILL NOT CHANGE NRC:			
Slip rolling	**SR** or **RD**	2	4
Non-sliping rolling	**NSR** or **RS**	1	5
Couplers	**Cou**	2	4
Joint's Motion	**M**	-1	-1
Forces	**Force**	3	6

Figure 42. Summary of “Table of the value of “f_i” (the nº of degrees of freedom) of some constraints”.

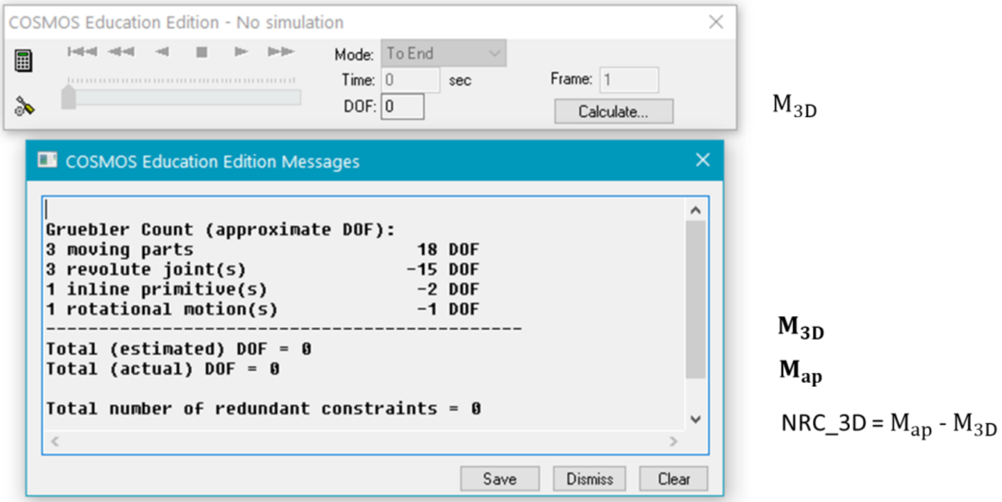
4.3. APARENT MOBILITY AND REDUNDANT CONSTRAINTS. HOW TO UNDERSTAND THEIR RELATION

When the mechanism is overconstrained, redundancies will appear. Redundancies usually appear when new constraints are defined but they do no change body/ies's motion, because they do not change body/ies' DOF or because (2) they add a constraint which conflicts with an existing constraint. Only non-redundant constraints must be defined.

The solver ignores/removes redundant constraints to achieve zero degrees of freedom, and then it generates a solution. However, if redundancies are not eliminated before performing a dynamic analysis, it may not get accurate values for measure connection reactions or load reactions. The outcome results may be incorrect, although the motion is correct. With the purpose of getting complete and accurate reaction forces, it is critical to eliminate redundancies from the mechanism. Alternatively, for kinematic problems, where only there is an interest in displacement, velocity or acceleration, redundancies do not affect the design and performance of the mechanism.

By default, the software calculates the DOF and redundancies for the model each time an analysis is made. Some CAE programs can automatically delete redundancies to get a simulation. So, by looking at the inform of the simulation, redundancies can be identified. It is a tool to keep self-aligning a mechanism.

It is useful to check the values of M_{3D}, M_{ap} and NRC calculated by the program. When using Recurdyn, it is needed to simulate a "Pre-Analysis Simulation" or a "Dynamic Simulation". Results will appear in the command windows, at the beginning of the inform. When using COSMOSMotion, "COSMOSMotion" tab > "Show Simulation Panel" > "Calculate". It is very useful to have that window opened. This way, every time a new constraint is created in the model, different aspects can be checked. But sometimes it is incorrect. So, to make sure it is correct, it is needed to run a simulation and go to 'Calculate' again. When this is done, it is sure that the numbers in the simulation panel will be correct.



Mode: To End
Time: 0 sec
DOF: 0
Frame: 1

```
|
Gruebler Count (approximate DOF):
3 moving parts                      18 DOF
3 revolute joint(s)                -15 DOF
1 inline primitive(s)               -2 DOF
1 rotational motion(s)              -1 DOF
---------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 43. Simulation panel of COSMOSMotion and the equation of each value.

M_{3D} is calculate by using this equation: $M_{3D} = 6 * (Nb - Nco - 1) + \sum_{i=1}^{i=Nco} f_i$. M_{ap} is calculated by internal equations. It just appears in the simulation panel. NRC = M_{ap} - M_{calc} gets updated when: (1) closing a chain; or (2) when changing a constraint in an already closed chain whose all joints have been already defined. The objective is to get NRC=0, so $M_{3D} = M_{ap}$. So, if NRC≠0 there is a not-self-aligned chain. In that chain:

- if NRC > 0, there are redundant constraints to be removed.
- if NRC < 0, there are not enough constraints. So, $M_{int} > 0$.

The reason of (1) is because two constraints are needed when adding the last piece to the mechanism. When adding another piece, only 1 constraint is added. It can be easily checked in a 4-bar.

Next, there are some other aspects to be considered when self-aligning.

- When a piece has one coupler, a R joint or C+IP must be defined if it rotates and its $M_{ap} = 1$.
  - $M_{ap} = 1 = M_{3D} = 6(2-1-1)+1_R*1 = 6(2-1-1)+2_C*1+5_{IP}*1$
- When a piece has two or more couplers, only a R can be defined if it rotates and its $M_{ap} = 1$
  - $M_{ap} = 1 = M_{3D} = 6(2-1-1)+1_R*1$. But it is not possible $M_{3D} = 6(2-2-1)+1_R*1+2_C*1=-3$
- When evaluating constraints as redundant constraints, it depends on the vectorial arrangement of the blocked DOF by a joint, as well as the arrangement of the bodies in the mechanism.
- If the CAE program uses M_{calc} as M_{3D}, when defining joint from a self-aligned 2D diagram, redundant constraints may will appear.
- Some problems related with the pieces themselves can appear.
  - Redundant pieces can exist. Usually, this problem can be solved by creating a fixed joint between them. It is the case of the following two green pieces:

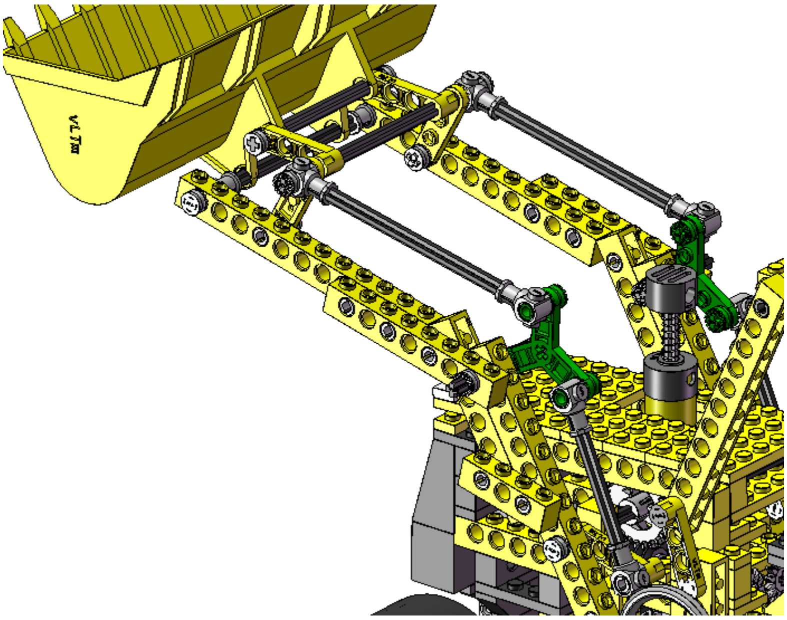
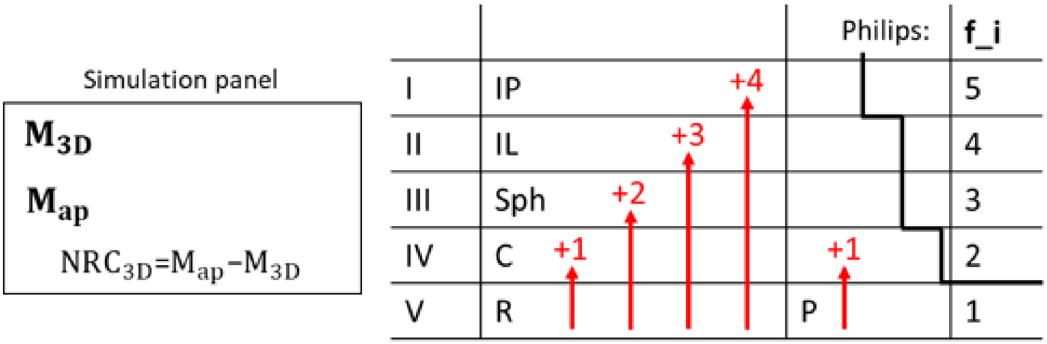


Figure 44. Both green pieces are redundant pieces. To get the LTM_8862's mechanism "frontal shovel" self-aligned, a fixed joint between both might have to be defined. This way, it becomes as only one piece.

  - If needed to self-align the mechanism, a new design of pieces or a new arrangement of pieces must be considered.
- Some interesting points about the path of self-aligning a mechanism in a CAE program:
  - A machine is self-aligned when all its mechanisms are self-aligned, and so when all kinematic chains are self-aligned. Before keep going whit the self-aligning process, chain by chain must be already self-aligned. This way, it is needed to check output values of M_{3D}, M_{ap} and NRC calculated by the program.
  - First, all joints of the kinematic chain are placed. For instance, only with R.
  - Secondly, once the chain is fully defined with constraints, they can be changed to self-align it. Also, other constraints can be added, for example motion in the joint.
  - When changing joint or contact constraints, in order to consider a changed constraint as valid, the M_{ap} before and after the change cannot change. If it does, previous constraint must be restored, and another different option must be tested. It might be needed to change more than one constraint. M_{3D} does can change.
  - After changing a constraint, NRC > 0 or < 0 number gets reduced or augmented, respectively, by the same nº of the result of changed joint's mobility minus the previous joint's mobility (f_i). Said number is the red arrow in next figure, which is a 'summary' of the "Table of Reshetov". Notice that in following table, a R joint can be changed to C, Sph, IL or IP joint, but a P joint only to a C joint



		Philips:	f_i
I	IP		5
II	IL		4
III	Sph		3
IV	C		2
V	R	P	1

Figure 45. How many degrees of freedom are augmented when going from a revolute joint to a C, Sph, IL or IP joint, and from a P joint to a C joint.

The following figure is going to be used as an example to explain how M_{ap} and NRC are related.

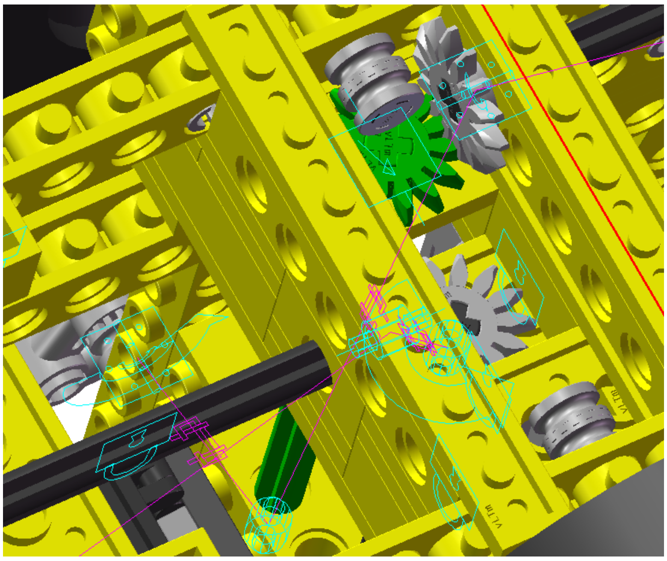


Figure 46. Auxiliar photo to explain different kinematic configurations.

The green piece 036 is jointed to the chassis 001 two times. 001 is considered as fixed to the ground, and 036 is a moving piece which only can move about its axis, so M_{ap} = 1 DOF. The sentence "if just one bearing were stablished, it would not be stable to be used" is not useful when self-aligning. Only non-redundant constraints must be defined. If two R joint were put, it would be overconstrained: $M_{3D} = 6*(2-2-1) + 1_R*2 = -4$ DOF. So, by changing some joints, it must augment 4 DOF. These two R joints are over-constraining the mechanism because they are blocking the same

- by changing both R joints by Sph + IL: $M_{3D} = 6*(2-2-1) + 3_{Sph}*1 + 4_{IL}*2 = +1$ DOF. Also, according to the number of the red lines in previous figure, -4 + 2 + 3 = +1. In this case, a coupler cannot be defined to simulate the gear, because there is no R or P or C joint. This configuration for this piece is not useful.
- by changing both R joints by C + IP: $M_{3D} = 6*(2-2-1) + 2_C*1 + 5_{IP}*2 = +1$ DOF. Also, according to the number of the red lines in previous figure, -4 + 1 + 4 = +1. In this case, a coupler can be defined to simulate the gear.
- by changing both R joints by one R: $M_{3D} = 6*(2-1-1) + 1_R*1 = +1$ DOF. In this case, a coupler can be defined to simulate the gear. This solution show why non-redundant have to be defined. Attending to the equation, it will over-constrain the mechanism; but in the physical mechanism, two R joints will be implemented.

- Imagining other case, it is interesting to notice that if this piece nº 036 had 3 joints with the 001, Map = 1 and $M_{3D} = 6*(2-3-1) + x$. By solving, x=13. It is possible with two IP and 1 Sph joint. But since there is a couple (a gear), it is needed one R, P or C joint. So, the maximum mobility is two IP and one C, that is to say, 5 + 5 + 2 =12. Thus, it is needed to see the other joints in the same kinematic chain of 036. Also, if one of these three joints are redundant can be analyzed. Sometimes, this constraint might will not be able to be removed: 12 – 13 = -1 DOF.

These two configurations, Sph + IL or C + IP, are kinematically equivalent to a R joint.

5. KINEMATIC CHAINS & KINEMATIC DIAGRAMS

5.1. ABOUT KINEMATIC DIAGRAMS

A kinematic diagram is a drawing where all the pieces and joints of the considered mechanism appear. It can be analyzed in two dimensions if it is a planar mechanism, so M_{2D} is calculated; or in three dimensions if it is a spatial mechanism, so M_{3D} is calculated. Here is the process to complete a kinematic diagram:

1. To identify the fixed to the ground piece and to identify the moving ones.
2. To identify the constraints.
3. To identify any points of interest.
4. To draw the kinematic diagram.
5. To calculate all about mobility.

In the kinematic diagram, the ground is not identified by a nº because it is not a piece, but by "Ground" The fixed to the ground piece is also called as base piece, because the movement of the rest of pieces is defined with respect to it, so it is usually identified as the 1st piece. It is represented with some small diagonal lines. Finally, moving parts are identified by 2, 3, 4, …. Only pieces are identified in the kinematic diagram. They appear within a circle:

Body's ID.

On the other hand, it is needed 3 rows for each constraint:

nº of identification of the constraint
f_i - letter of the constraint
First body - Second body

Here are some useful icons when drawing the diagram by hand:

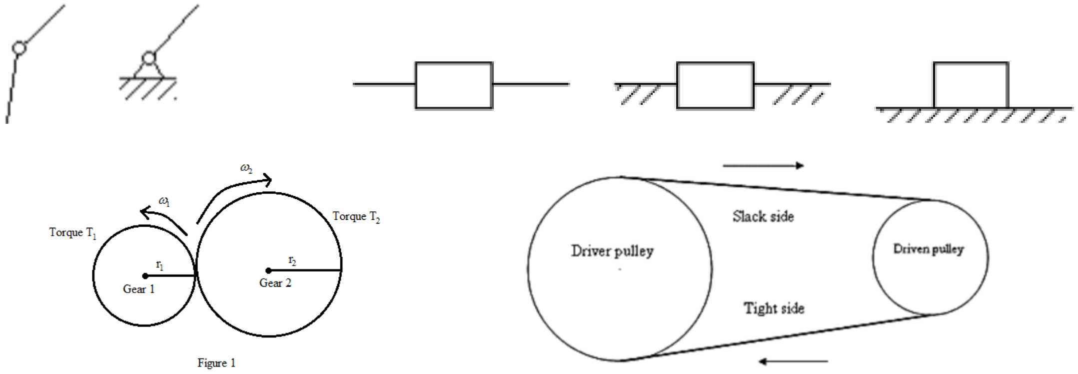


Figure 47. Icon of a R joint, a P joint, a gear, and a chain/belt coupler to be used in a planar kinematic diagram. Source: https://www.cs.cmu.edu/~rapidproto/mechanisms/tablecontents.html

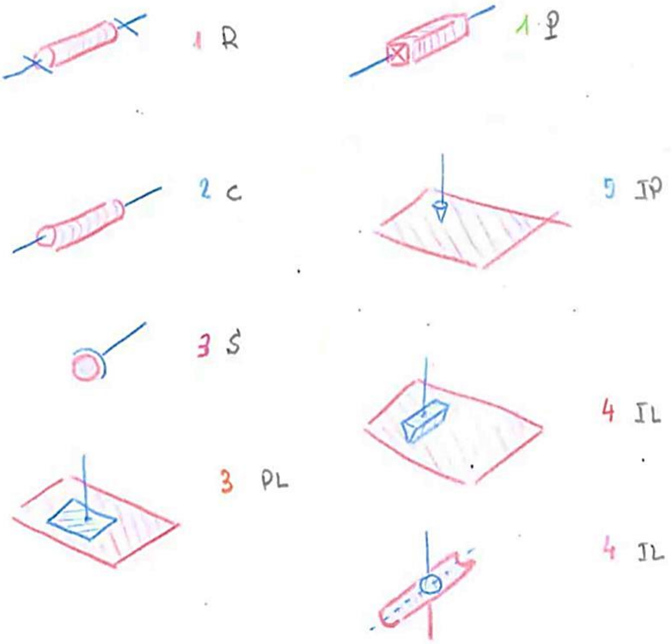
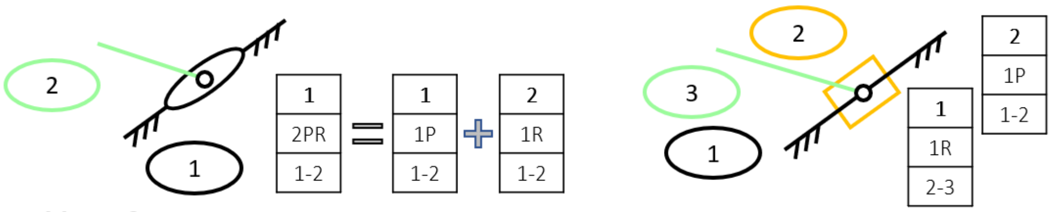


Figure 48. Icons of the joints in the Table of Reshetov to be used in manual spatial kinematic diagrams. Source: http://www.upv.es/vltmodels/index.html

Next, there are some aspects about the kinematic diagrams.

- When in a 'place' meets a few pieces, the nº of joints to consider is the nº of pieces which meet minus 1. For example, looking again at said image, is top left point from fixed part, it can be observed that there are two joints. Joint nº 17 and 10 are related with piece nº 12, 7 and 1. Thus, nº joints to consider = nº of bodies in the meeting – 1
- Although it has not found to be proven, usually there are – (Nb – Nco – 1) chains.
- The value of M_{3D} and M_{ap} before taking into consideration the pneumatic cylinders have to be the same as after doing so. In addition, it has to be mentioned that non-repeated linear actuators (pneumatic cylinders, shock absorbers, …) add +1 DOF to the M_{ap} of the mechanism. 'Non-repeated' means that their movement is non-dependent. If there were three linear actuators in a mechanism but one of them were repeated, the M_{ap} of that mechanism would be 2.
- A "pin-in-a-slot" - **slot pin, SP**, or "*pasador ranura", **PR***- removes 1 body and the same motion takes place.



Map = 2
M2D = 3(2-2-1)+2 = 2

2 bodies & 1 joint are needed

Map = 2
M2D = 3(3-3-1)+2 = 2

3 bodies & 2 joints are needed

Figure 49. Kinematic diagram of a slot pin.

- When in a chain there are only P joints, it is needed to add 1 piece and 1 R. For example, the piece nº 6 in the left image can be divided in two: piece nº 4 and nº 3 in the right image. To join the, a R joint must appear. Now, each 'new' piece moves along one axis, so two P joints appear (nº 4 and nº 5).

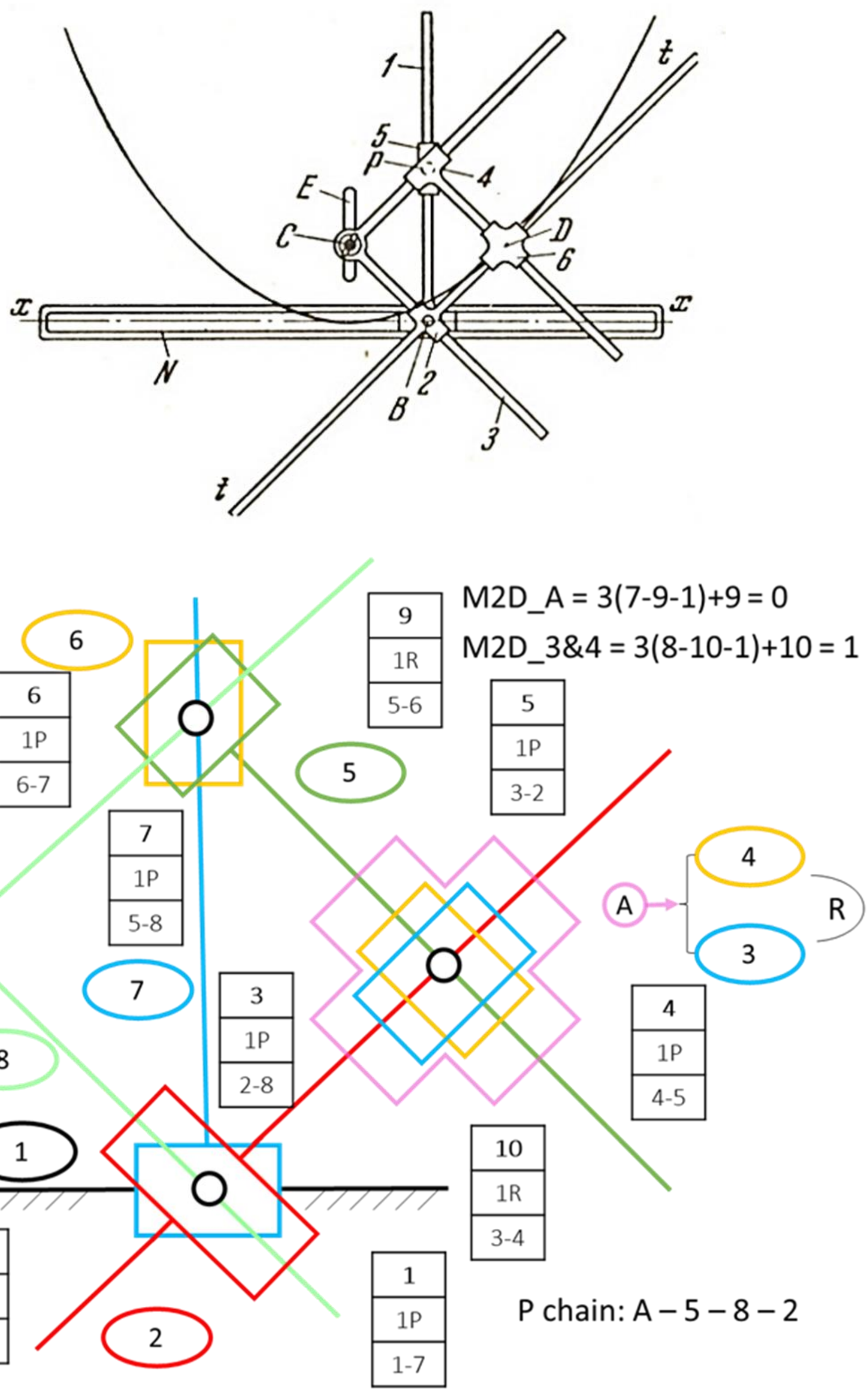
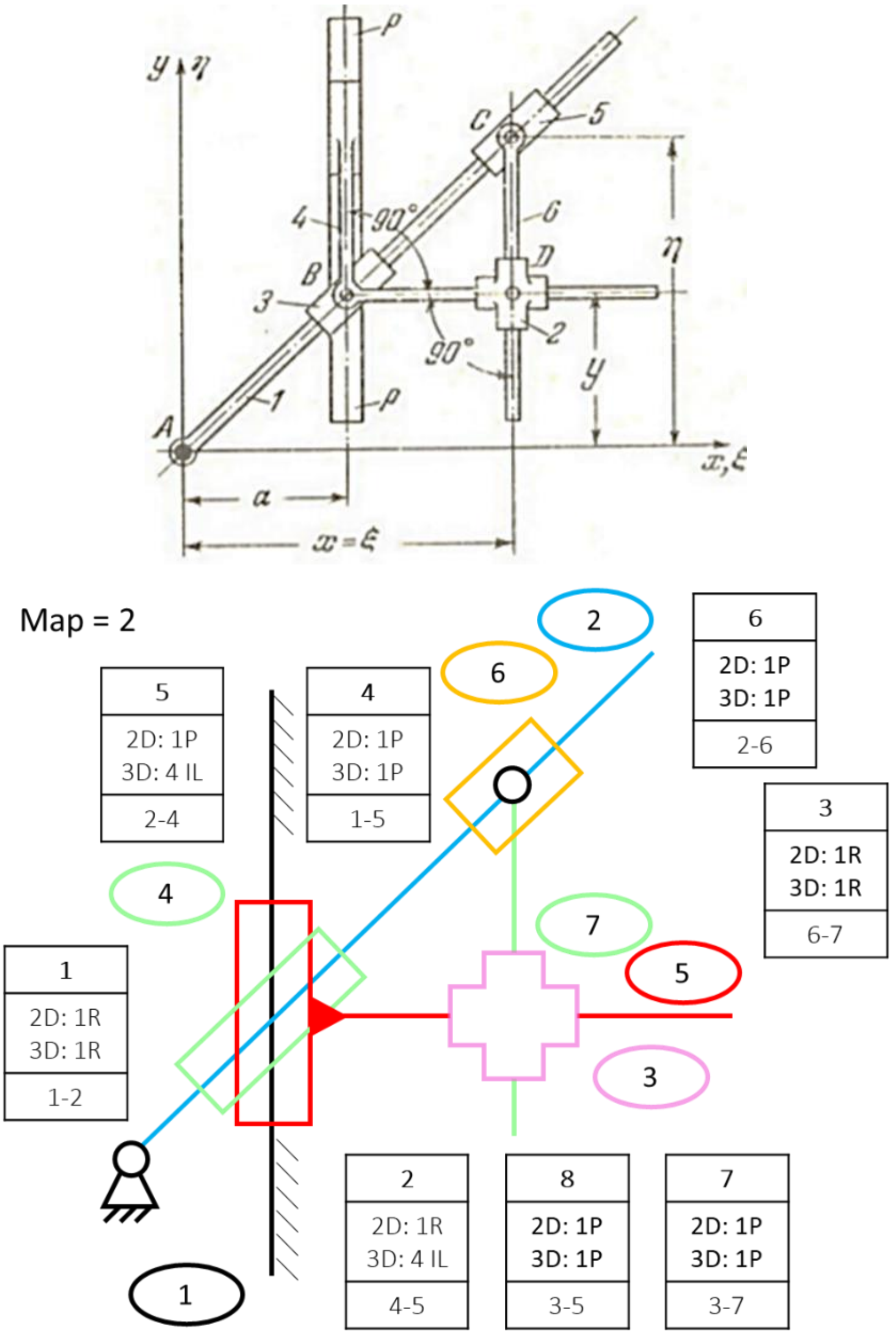


Figure 50. Kinematic diagram of the planar mechanism "Artobolesky 1111". A kinematic chain of only P joints is identified. Source: Artobolevsky, I. I. (1976). "Mecanismo en la tecnica moderna. Tomo I".

5.2. SELF-ALIGNING A PLANAR MECHANISM ACCORDING TO ITS KINEMATIC CHAINS.

Next, mechanism “Artobolesky 1202” is going to be used to explain how to self-align looking at the kinematic chains. The same reasoning can be used when self-aligning 3D mechanisms. Just to highlight that all user-defined constraints must be non-redundant constraints



M2D = 3(7-8-1)+1R*3+1P*5 = 2

M3D = 6(7-8-1)+1R*2+1P*4+4 IL*2 = 2

Figure 51. Kinematic diagram of the planar mechanism “Artobolesky 1202”. Source: Artobolevsky, I. I. (1976). “Mecanismo en la tecnica moderna. Tomo I”.

1. M_{ap}

M_{ap} = 2, because it is the DOF that must be removed (blocked, constrained) in order to make any point of any body of the mechanism not to move.

2. M_{calc}

It is a planar mechanism, so $M_{calc} = M_{2D} = 3(7\text{-}J_{one}\text{-}1) + J_{one}$

3. NRC

To self-align any mechanism, it is needed NRC_3D = 0; but, since this mechanism is two dimensional, it is also needed $NRC_{23} = M_{2D} – M_{3D} = 0$. Thus, $M_{ap} = M_{2D} = M_{3D}$ if self-aligned.

3.1. NRC_2D = 0 (only if planar mechanism)

Since it is a planar mechanism, it is wanted to FIRST design in a way that $NRC_{2D} = 0$ -if it were a spatial mechanism, it would not make sense this step-. So, it is calculated how many types and nº of joints are needed to self-align in 2D -it also could be done in 3D: NRC_{3D}= 0-. Solving, J_{one} = 8, that is to say, 8 joints of one DOF are needed to self-align the mechanism. Creating J_{one} joints as follows in the diagram, it will be self-aligned in 2D: M_{2D} = 3(7-8-1) + 1R*3 + 1P*5 = 2.

3.2. NRC_23 (only if planar mechanism)

Once there are some constraints in the diagram, it is wanted to self-align in 3D. If it were a spatial mechanism, joints would have to be created ‘randomly’. Now, M_{3D} can be calculated:

M_{3D} = 6(7-8-1) + 1R*3 + 1P*5 = -4.

So, NRC_23 = M_{2D} – M_{3D} = +6 redundant constrictions. As it is NRC > 0, it must augment +6 DOF between all chains of figure’s mechanism. In addition. It is already known that all chains must be self-aligned.

4. Identifying the kinematic chains

It is needed to self-align chain by chain. It might there be –(Nb – Nco – 1) kinematic chains.

In this case, it does there be –(Nb – Nco – 1) = -(7-8-1) = 2 chains: 1-2-4-5-1 (chain #1) and 4-5-3-7-6-2-4 (chain #2). Let “Pre” and “Post” be as follows. Pre: all the chains which has already been self-aligned. Post: all the chains which has not been self-aligned yet. So far, Pre = {empty]. Post = {#1; #2}.

4.1. Chain #1

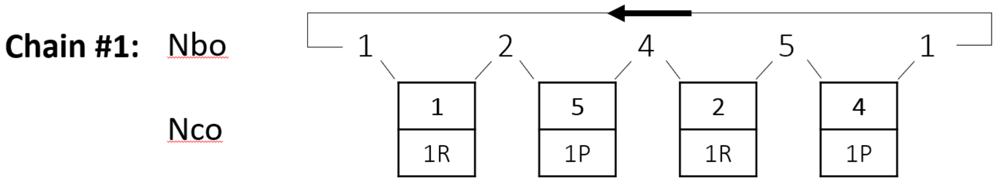


Figure 52. Chain #1 of “Artobolesky 1202”.

$M_{2D}^{\text{chain 1}}$ = 3(4-4-1) + 1R*2+1P*2 = 1

$M_{3D}^{\text{chain 1}}$ = 6(4-4-1) + 1R*2+1P*2 = -2

$\mathrm{NRC}_{23}^{\text{chain 1}} = \mathrm{M}_{2\mathrm{D}}^{\text{chain 1}} - \mathrm{M}_{3\mathrm{D}}^{\text{chain 1}} = +3$

So, in chain #1, it must be augmented +3 DOF. To do so, for example, a R joint can be changed by an IL joint. By doing so, chain #1 will be self-aligned. So, joint 2 will not be any more a R joint but a IL joint. Thus:

$\mathrm{M}_{2\mathrm{D}}^{\text{chain 1}\,\prime}$ = 3(4-4-1) + 1R*2+1P*2 = 1

$\mathrm{M}_{3\mathrm{D}}^{\text{chain 1}\,\prime}$ = 6(4-4-1) + 1R*2+1P*2 = 1

$\mathrm{NRC}_{23}^{\text{chain 1}\,\prime} = \mathrm{M}_{2\mathrm{D}}^{\text{chain 1}} - \mathrm{M}_{3\mathrm{D}}^{\text{chain 1}} = 0$

Since NRC = 0, chain #1 is self-aligned. So far, Pre = {#1] and Post = {#2}.

4.2. In next chains…

Now, there are NRC_{23} - $\mathrm{NRC}_{23}^{\text{chain 1}\,\prime}$ = +3 DOF left to be augmented in all Post chains. In addition, any of the already modified joints in Pre chains cannot be modified in Post chains because If they were modified again, the $\mathrm{M}_{3\mathrm{D}}$ would be modified too. Thus, in Post chains, the already modified joints in Pre chains are supposed not to be modified again. In addition, when calculating the mobility in Post chains, original joints must be used. To sum up already modified joints in Pre chain are not updated nor changeable when self-aligning Post chains.

In this example, joint 2 will not be modified in current Post chains (#2): it will be still considered as R joint, as before being changed in current Pre chains (#1).

4.3. Chain #2

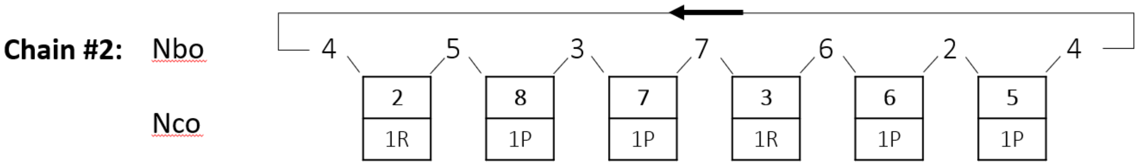


Figure 53. Chain #2 of "Artobolesky 1202".

$\mathrm{M}_{2\mathrm{D}}^{\text{chain 2}}$ = 3(6-6-1) + 1R*2+1P*4 = 3

$\mathrm{M}_{3\mathrm{D}}^{\text{chain 2}}$ = 6(6-6-1) + 1R*2+1P*4 = 0

$\mathrm{NRC}_{23}^{\text{chain 2}} = \mathrm{M}_{2\mathrm{D}}^{\text{chain 1}} - \mathrm{M}_{3\mathrm{D}}^{\text{chain 1}} = +3$

So, in chain #2, it must get +3 DOF. To do so, for example, a P joint can be changed by an IL joint. By doing so, chain #2 will be self-aligned. So, joint 5 will not be any more a P joint but a IL joint. Thus:

$\mathrm{M}_{2\mathrm{D}}^{\text{chain 2}\,\prime}$ = 3(6-6-1) + 1R*2+1P*2 = 1

$\mathrm{M}_{3\mathrm{D}}^{\text{chain 2}\,\prime}$ = 6(6-6-1) + 1R*2+1P*1+ 4 IL*1= 1

$\mathrm{NRC}_{23}^{\text{chain 2}\,\prime} = \mathrm{M}_{2\mathrm{D}}^{\text{chain 2}} - \mathrm{M}_{3\mathrm{D}}^{\text{chain 2}} = 0$. So far, Pre = {1; 2]. Post = {empty}.

4.4. In next chains…

Now, there are NRC_{23} – ($NRC_{23}^{\text{chain 1}}\,'$ + $NRC_{23}^{\text{chain 2}}\,'$) = 0 DOF left to be augmented in all non-already self-aligned chains. In addition, so far in this example,

- joint 2 will not be modified in 'Post' chains (#2): it will be still considered as R joint, as before being changed in chain #1.
- joint 5 will not be modified in 'Post' chains (empty): it will be still considered as P joint, as before being changed in chain #2.

4.5. There are no more chains. Results

Since all chains has been already self-aligned, because NRC_{23} – ($NRC_{23}^{\text{chain 1}}\,'$ + $NRC_{23}^{\text{chain 2}}\,'$) = 0 DOF, there are no chains left to be self-aligned. These results must appear in the diagram. That is why "diagram" has been considered as synonym of "kinematic diagram with all self-aligned mechanism's mobilities".

Finally, this is the self-aligned considered mechanism:

$M_{ap}\,'$ = 2

$M_{2D}\,'$ = $M_{2D}^{\text{chain 1}}\,'$ + $M_{2D}^{\text{chain 2}}\,'$ = 2

$M_{3D}\,'$ = $M_{3D}^{\text{chain 1}}\,'$ + $M_{3D}^{\text{chain 2}}\,'$ = 2

$NRC_{2D}\,'$ = $M_{ap}\,'$ – $M_{2D}\,'$ = 0

$NRC_{3D}\,'$ = $M_{ap}\,'$ – $M_{3D}\,'$ = 0

$NRC_{23}\,'$ = $M_{2D}\,'$ – $M_{3D}\,'$ = 0

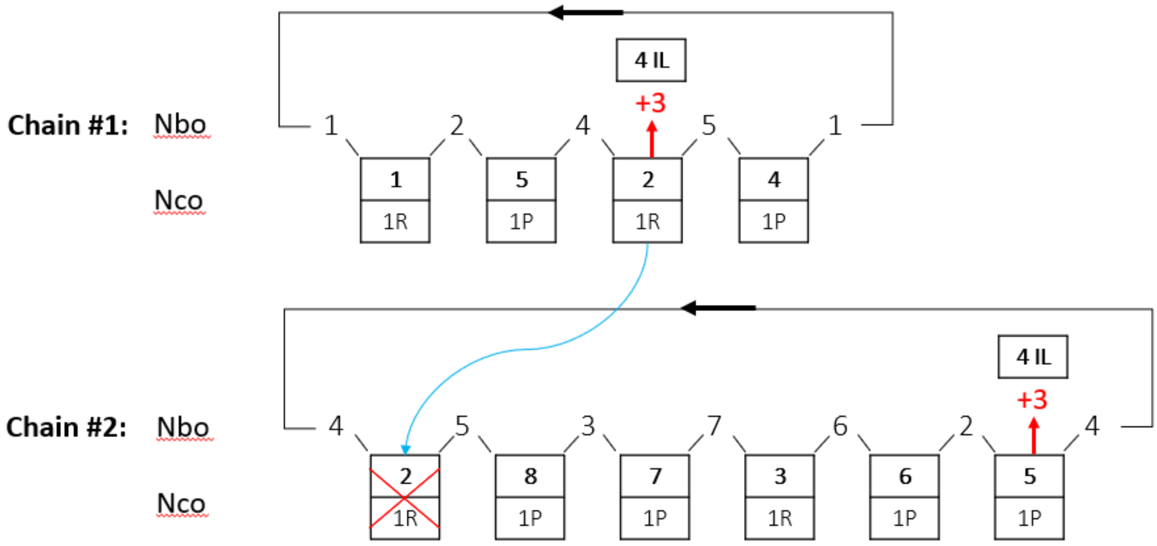


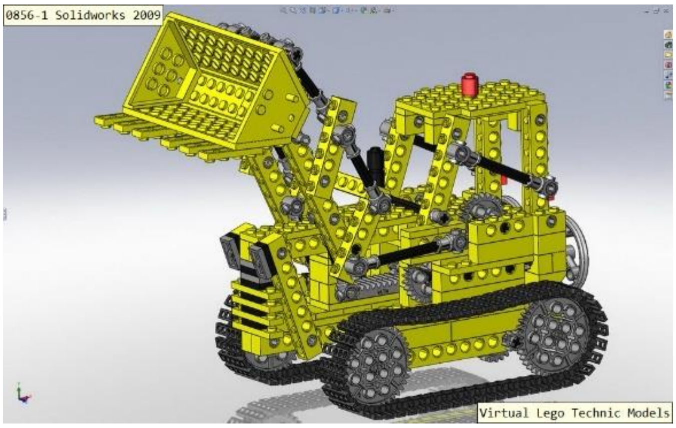
Figure 54. Summary of the self-aligning process of "Artobolesky 1202".

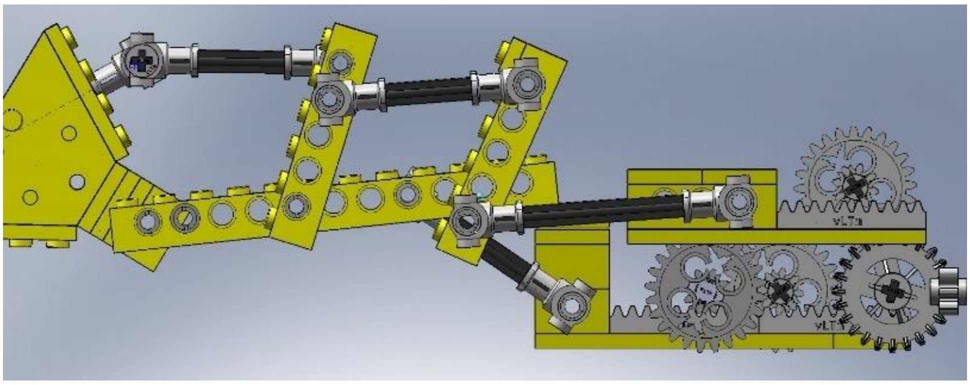
5. Motion constraints

Once the mechanism is self-aligned, the purpose of defining motion constraints is to go from a M_{calc} and M_{ap} > 0 to = 0. It is possible because, when adding 1 motion in the properties of the joint, M_{calc} and M_{ap} are reduced by 1. So, mechanism runs as wanted. It will be needed to control as many inputs (motion) as M_{ap}, for example with an electrical engine. Thus, the simulation represents the reality.

5.3. EXAMPLES OF KINEMATIC DIAGRAMS

5.3.1. M2D - EXAMPLE 1 - VLTM-0856-1





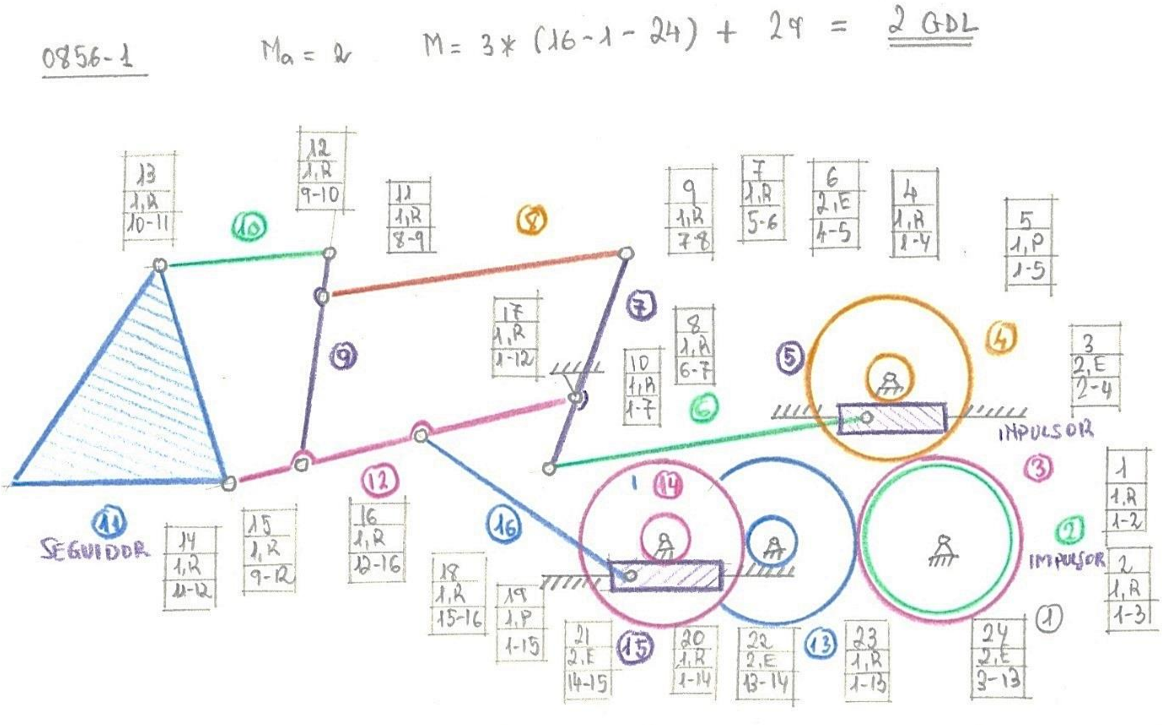
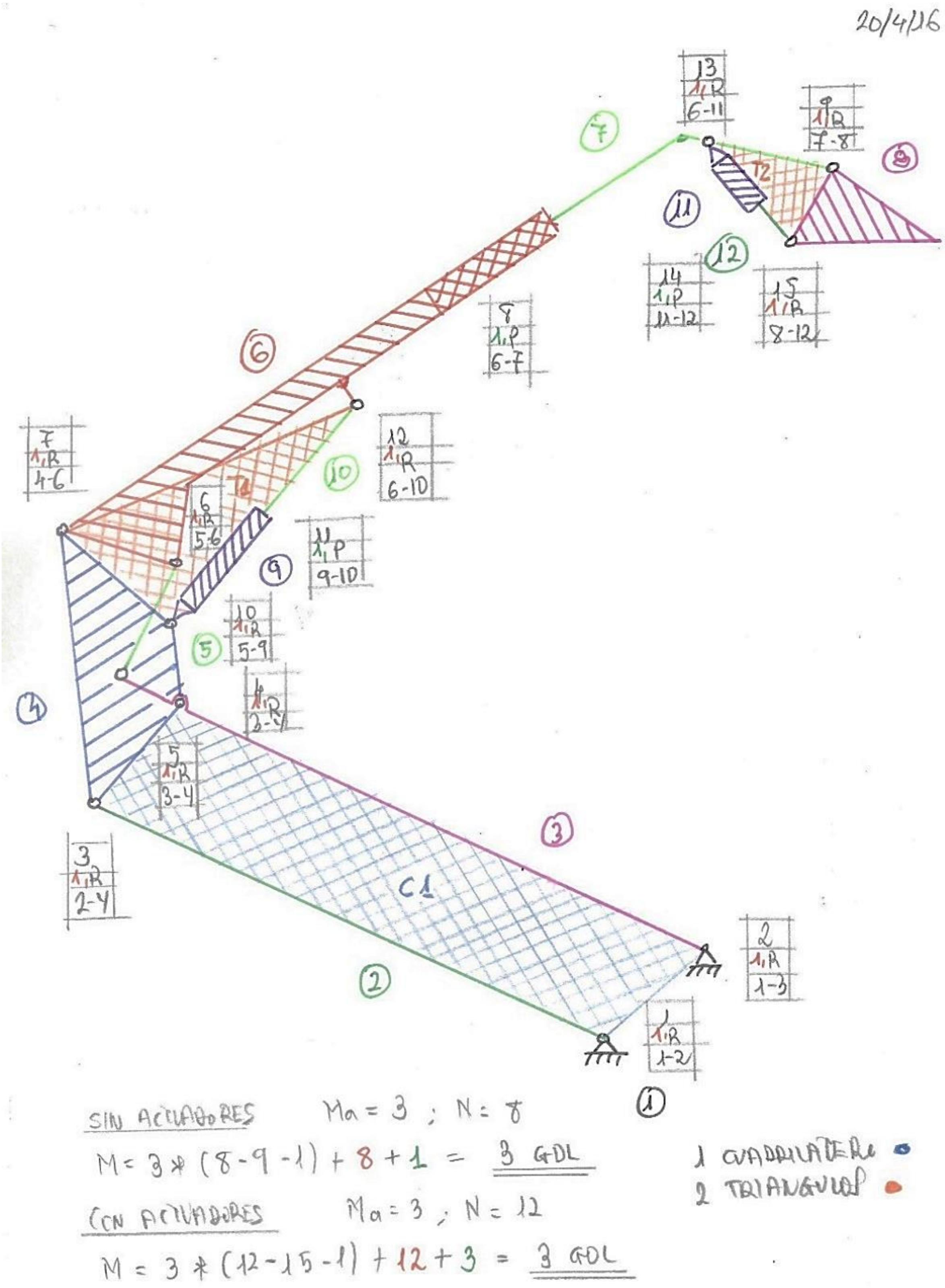


Figure 55. A planar mechanism and its kinematic diagram of the LEGO® Technic model (LTm) 0856-1. Source: http://www.upv.es/vltmodels/10-c33-obras-pub-r.html.

5.3.2. M2D - EXAMPLE 2 - JLG-T350-2D

Figure 56. A machine to work at height: T350 elevator. Source: http://www.upv.es/vltmodels/10-c33-obras-pub-r.html.



SIN ACTUADORES   Ma = 3 ; N = 8

$M = 3 * (8 - 9 - 1) + 8 + 1 = \underline{3\ GDL}$

CON ACTUADORES   Ma = 3 ; N = 12

$M = 3 * (12 - 15 - 1) + 12 + 3 = \underline{3\ GDL}$

1 CUADRILATERO ●

2 TRIANGULOS ●

Figure 57. A planar mechanism and its kinematic diagram of a machine to work at height. Source: http://www.upv.es/vltmodels/10-c33-obras-pub-r.html.

5.3.3. M_{3D} - EXAMPLE 1 - THANG-DUMPTRUCK2INV

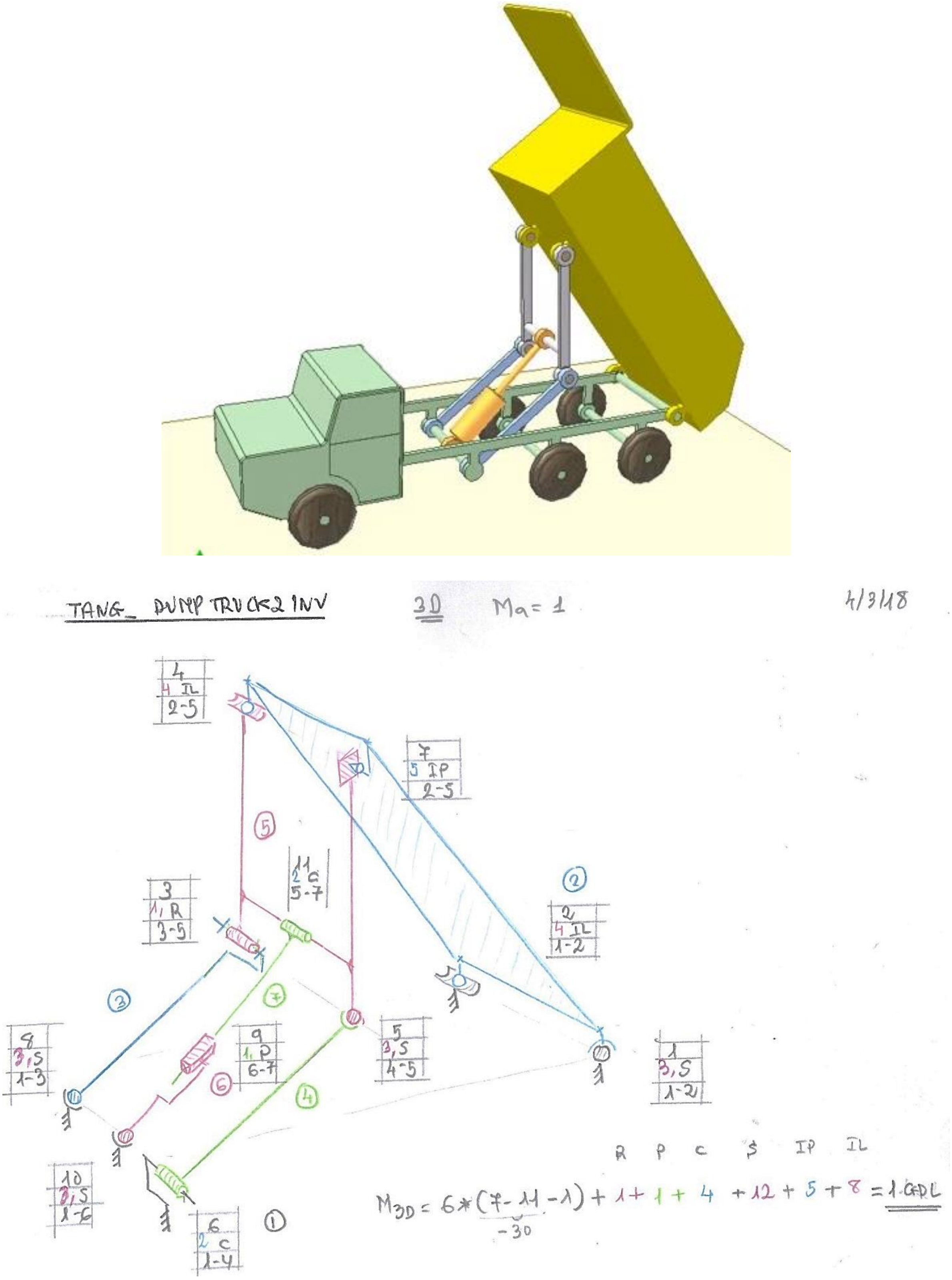


Figure 58. A spatial mechanism and its kinematic diagram of a dump truck. Source: http://www.upv.es/vltmodels/index.html.

5.4. EXAMPLES OF JOINTS IN A REAL MACHINE

Figure 59. An example of some joints in a real machine.

6. PHILIPS' POINTS. OTHER JOINTS

When it is not a common bearing or it is any different joint to described previously, a very ingenus method can be used. It has been called "Philips' points". It has been found that some nº of preferential erosion points appears. The nº of points coincides with the nº of DOF reduced/blocked by the joint. So, one point is equal to a joint with f_i = one, that is to say, the nº of points coincides with the mobility/connectivity of that joint. For example, the fixed joint lets no movement, so there is 0 points of erosion. So 'fixed' joint restricts 6 DOF because lets 6 – 6 = 0 DOF.

Thus,

f_i = 6 – nº of preferential erosion points appears for joint nº 'i'

$$M_{3D}^{\text{by Philips}} = \begin{cases} 6 - \sum_{i=1}^{i=Nco} f_i \text{ , if just 2 bodies} \\ 6 * (Nb - Nco - 1) + \sum_{i=1}^{i=Nco} f\_i \text{ , if 3 or more bodies} \end{cases}$$

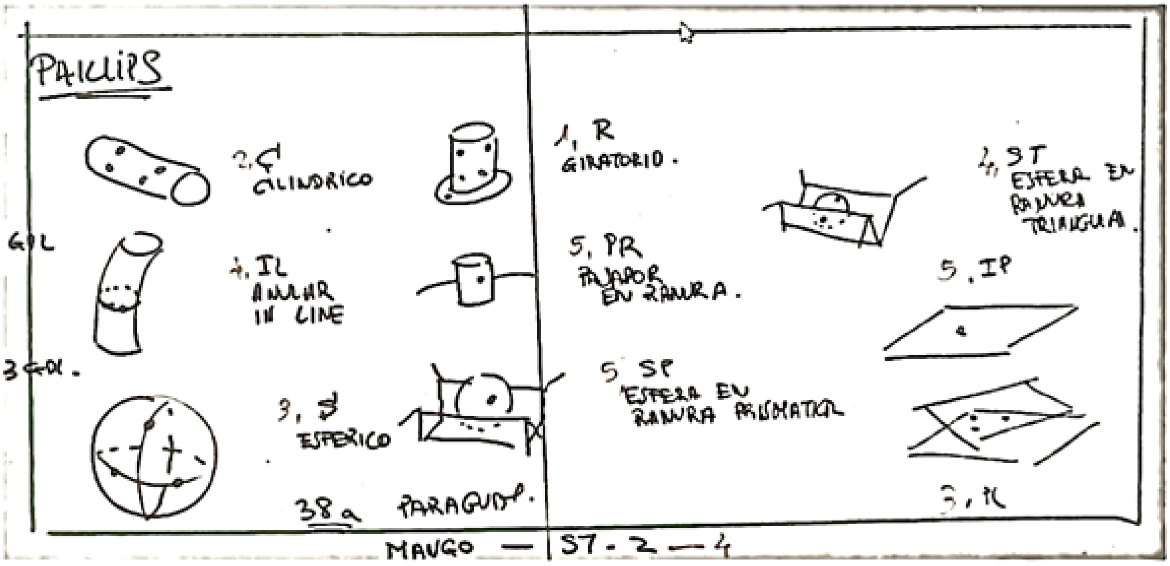
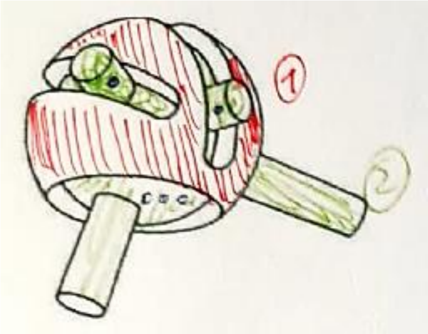
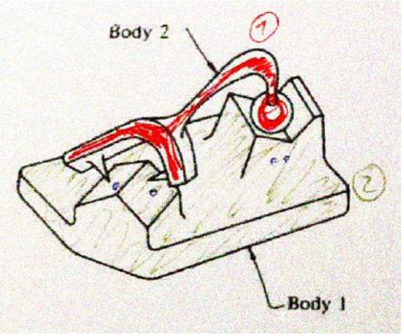


Figure 60. Some of the joints with Philip's points. Source: http://www.upv.es/vltmodels/09-c28-phillips.html.

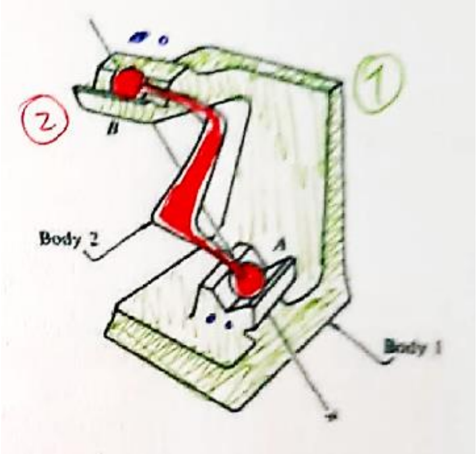
Here are some examples. In the last one, there are three bodies: 2 moving parts (1 umbrella, nº 1, and 1 container, nº 3) and the fixed to the ground part (the walls and floor, nº 2).



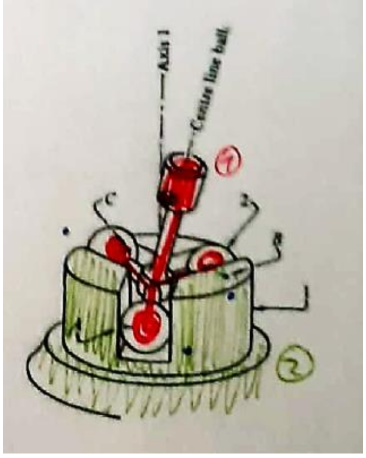
$$M_{3D}^{\text{by Philips}} = 6 - (1_{PR} + 1_{PR} + 3_{Sph}) = 1$$



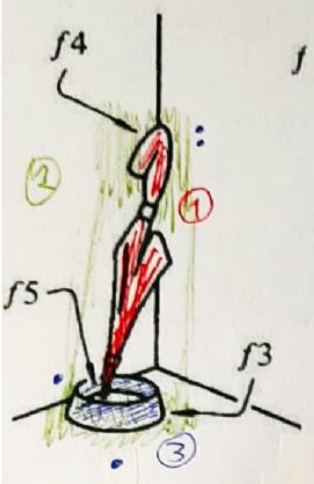
$$M_{3D}^{\text{by Philips}} = 6 - (2_{ST} + 1_{IP} + 1_{IP}) = 2$$



$$M_{3D}^{\text{by Philips}} = 6 - (2_{ST} + 1_{IP} + 1_{IP}) = 2$$



$$M_{3D}^{\text{by Philips}} = 6 - (1_{SP} + 1_{SP} + 1_{SP}) = 3$$



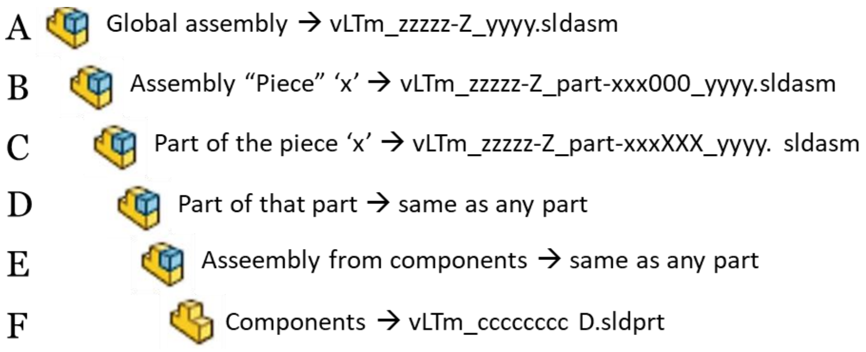
$$M_{3D}^{\text{by Philips}} = 6 * (3 - 3 - 1) + (4_{ST,\ \text{hand}} + 5_{IP,\ \text{vertex}} + 5_{PL,\ \text{recipient}}) = 1$$

Figure 61. Some examples of Philip's points method. Source: http://www.upv.es/vltmodels/09-c28-phillips.html.

7. METHODOLOGY USED TO CREATE THE SIMULATED MODELS

7.1. STEP 1. TO ASSEMBLY THE VEHICLE IN PHYSICAL AND IN A CAD PROGRAM. SOLIDWORKS® 2020

First phase consists in creating a virtual model of the machine in a CAD program. To help doing this, the corresponding model has bought online, and its manual has been used as a guide to assembly the model (the guides can be found online too). To virtualize the models, in this case, it has been used SolidWorks® 2020. It is very important to use a proper nomenclature when naming the files. **If two pieces are wanted to have relative movement, they must be in different files**. Each model or global assembly is a group of assemblies and parts. The nomenclature is as follows:



A Global assembly → vLTm_zzzzz-Z_yyyy.sldasm
B Assembly “Piece” ‘x’ → vLTm_zzzzz-Z_part-xxx000_yyyy.sldasm
C Part of the piece ‘x’ → vLTm_zzzzz-Z_part-xxxXXX_yyyy. sldasm
D Part of that part → same as any part
E Asseembly from components → same as any part
F Components → vLTm_cccccccc D.sldprt

zzzzz: code of the model from LEGO®
Z: nº of assembly of that model, 1 or 2
yyyy: when global assembly is created, year

xxx: nº of part
XXX: nº of part of the part ‘xxx’
cccc: id. of that component
D: description of the component ‘cccc’

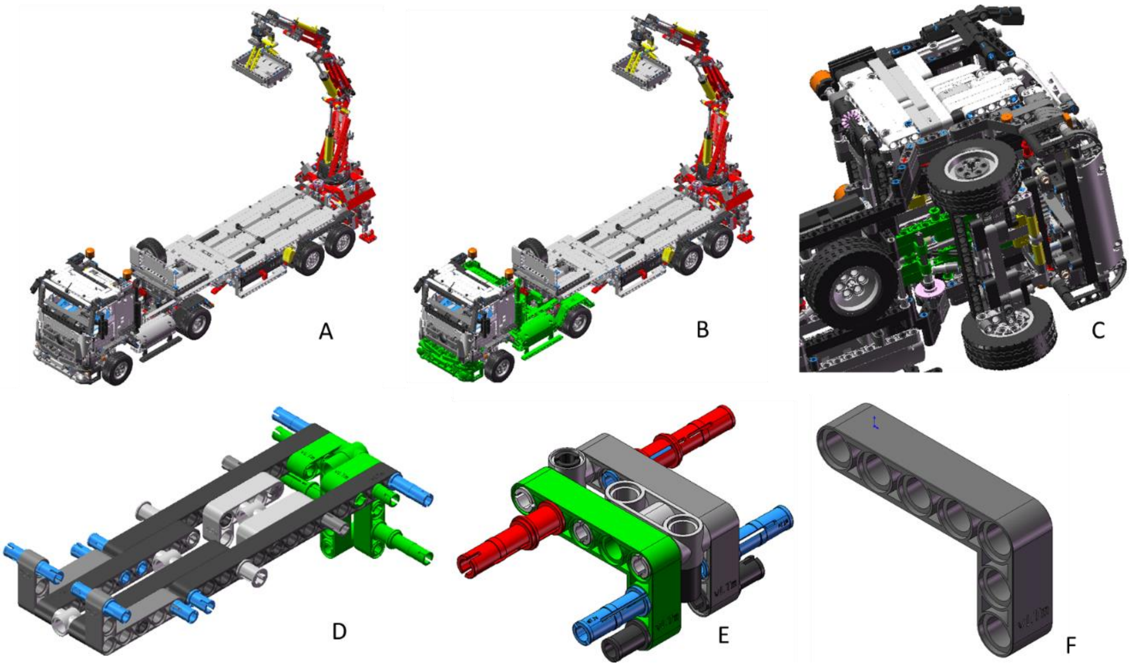


Figure 62. Nomenclature of CAD files used in the Thesis.

It is interesting to indicate what part and assembly is for SolidWorks®:

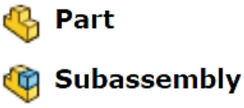


Figure 63. Icon of part and assembly in SolidWorks® 2020.

It is recommended that each assembly from component does not have more than ten components.

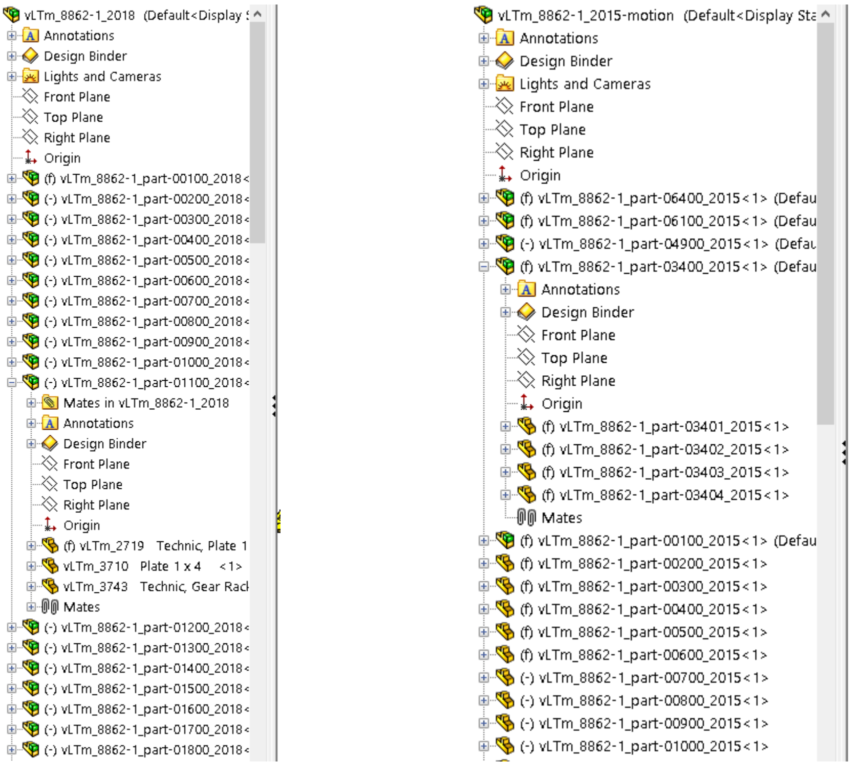
Once the CAD file is finally finished, it is recommended to fix all pieces (parts & assemblies). When opening a *.x_b file, pieces are placed according to its position with respect to the origin of the original file.

It can be very useful to create different files by copy-pasting, if important changes are going to be made in the file; also, to name it with a descriptive name.

7.2. STEP 2. REDUCING Nº OF “ORIGINAL” CAD FILES

To reduce the nº of equations the program must manage, it is recommended to save pieces which have relative movement with respect to other pieces as a *.sldprt. It can be done in the same SolidWorks® program. It must be done for all the pieces in the model. This process has been called as “**compacted**” by the supervisor of the Thesis.

This helps to run better the CAE program in next steps. However, once this process is finished, assemblies do not have parts. All previous parts which conformed the assembly are converted into one part. Only the assemblies which had assemblies, will be now an assembly.



```
vLTm_8862-1_2018  (Default<Display S
  Annotations
  Design Binder
  Lights and Cameras
  Front Plane
  Top Plane
  Right Plane
  Origin
  (f) vLTm_8862-1_part-00100_2018<
  (-) vLTm_8862-1_part-00200_2018<
  (-) vLTm_8862-1_part-00300_2018<
  (-) vLTm_8862-1_part-00400_2018<
  (-) vLTm_8862-1_part-00500_2018<
  (-) vLTm_8862-1_part-00600_2018<
  (-) vLTm_8862-1_part-00700_2018<
  (-) vLTm_8862-1_part-00800_2018<
  (-) vLTm_8862-1_part-00900_2018<
  (-) vLTm_8862-1_part-01000_2018<
  (-) vLTm_8862-1_part-01100_2018<
    Mates in vLTm_8862-1_2018
    Annotations
    Design Binder
    Front Plane
    Top Plane
    Right Plane
    Origin
    (f) vLTm_2719   Technic, Plate 1
    vLTm_3710   Plate 1 x 4     <1>
    vLTm_3743   Technic, Gear Rack
    Mates
  (-) vLTm_8862-1_part-01200_2018<
  (-) vLTm_8862-1_part-01300_2018<
  (-) vLTm_8862-1_part-01400_2018<
  (-) vLTm_8862-1_part-01500_2018<
  (-) vLTm_8862-1_part-01600_2018<
  (-) vLTm_8862-1_part-01700_2018<
  (-) vLTm_8862-1_part-01800_2018<
```

```
vLTm_8862-1_2015-motion  (Default<Display Sta
  Annotations
  Design Binder
  Lights and Cameras
  Front Plane
  Top Plane
  Right Plane
  Origin
  (f) vLTm_8862-1_part-06400_2015<1> (Defau
  (f) vLTm_8862-1_part-06100_2015<1> (Defau
  (-) vLTm_8862-1_part-04900_2015<1> (Defau
  (f) vLTm_8862-1_part-03400_2015<1> (Defau
    Annotations
    Design Binder
    Front Plane
    Top Plane
    Right Plane
    Origin
    (f) vLTm_8862-1_part-03401_2015<1>
    (f) vLTm_8862-1_part-03402_2015<1>
    (f) vLTm_8862-1_part-03403_2015<1>
    (f) vLTm_8862-1_part-03404_2015<1>
    Mates
  (f) vLTm_8862-1_part-00100_2015<1> (Defau
  (f) vLTm_8862-1_part-00200_2015<1>
  (f) vLTm_8862-1_part-00300_2015<1>
  (f) vLTm_8862-1_part-00400_2015<1>
  (f) vLTm_8862-1_part-00500_2015<1>
  (f) vLTm_8862-1_part-00600_2015<1>
  (-) vLTm_8862-1_part-00700_2015<1>
  (-) vLTm_8862-1_part-00800_2015<1>
  (-) vLTm_8862-1_part-00900_2015<1>
  (-) vLTm_8862-1_part-01000_2015<1>
```

Figure 64. Parts and assemblies before (left) and after (right) going from an after SolidWorks® 2007 to a SolidWorks® 2007 version CAD file.

In addition, if an after 2007 SolidWorks® version has been used to create the CAD file, it is needed to follow some steps in order to open it in SolidWorks® 2007. By doing this, all assemblies from recent versions of SolidWorks® can be opened in 2007 version.

It can be found here: https://www.youtube.com/watch?v=S6geQqcgv5s&ab_channel=RenderLover. It is just an option. Next, there is a brief transcription of the steps:

1. To open new SolidWorks® version (NSWV). To save the CAD file as *.step. Close NSWV.
2. To open the older SolidWorks® version (OSWV). To open the file created in step 1. To accept to run import diagnostic. Click yes in "Import Diagnostics". To save as *x_t. To close *.step.
3. To open the new *x_t file. To accept to run import diagnostic. Click yes in "Import Diagnostics". To accept to proceed with feature recognition [If this feature recognition is not asked by the program, to active FeatureWorks add-in and then click the option Insert > FeatureWorks > Recognize features]. To activate all automatic features (extrudes, volume, drafts …). To click okey to finish FeatureWorks operation.
4. That is all.

7.3. STEP 3. TO MOVE FROM THE FINAL SCALED CAD FILE TO A CAE FILE

The CAD filed can be copy-pasted, saved with other name, opened and finally no-needed parts can be suppressed. For instance, only parts from a mechanism can be no suppressed. Now, when saving as *x_b, only no-suppressed pieces will appear. This way, these files can be imported into the CAE program to only self-align that mechanism. So, once it is finally constrained, next mechanism can be imported. It lets to keep the evolution of M_{3D}, M_{ap} and NRC during the self-aligning process of one mechanism. It must be done with all pieces and mechanisms of the model.

The CAD file must be saved with an extension let by the CAE program. In this Thesis, two CAE programs have been used. First program is COSMOSMotion. It is just an add-in of SolidWorks® 2007. This way, to save as *.x_b is not needed. In the same opened CAD file using SolidWorks® 2007, all options to self-align the mechanism are activated by activating the COSMOSMotion add-in. Second program is Recurdyn. In this case, the *.sldasm file can be saved as *.x_b from SolidWorks®. Once Recurdyn is opened, this *.x_b file must be imported. When opened a CAE file, after clicking to edit a part, said part can be exported as *.rdbd to be used in another CAE file by importing it.

If the RecurDyn's toolkit "Chains" is needed to be used, once this toolkit is opened, *x_b must be imported. If it is not opened, the toolkit will not recognize the imported pieces. So, to define chain by using this part will not be possible. Once the file is imported, a "Subsystem" will be automatically created. To edit it, to go to database panel > subsystem > right click in the subsystem > edit. Also, the simulation can be initialized inside the subsystem (after clicking on edit it) or outside the subsystem. However, to use this toolkit is not very automatic. Flanges needs to perfectly fit within links. Because of this, it is not immediate to change the dimensions of sprockets and flanges. To solve this, before importing the *x_b into Recurdyn, it is useful to import only parts of the chain system. So, the sprocket and flange is defined using the toolkit, a scale has to be determined to fit imported files with sprocket and flange created with the toolkit. Once the scale is defined, it must be implemented in the CAD files of the part. But before, in the model's CAD file, mates must be suppressed; later, the scale must be defined in the parts; finally, mates must be created again. Now, the model is scaled .

This file can be imported into Recurdyn. The pieces from the CAD file and the pieces created using the RecurDyn's toolkit "Chains" fit. In addition, all middle plans of all flanges and sprockets need to be in the same plan. This a point to consider when assembling in the CAD file. Some auxiliar cylinders can be extruded from a common plan.

Figure 65. Predetermined single flange (left) and sprocket (right) in RecurDyn's toolkit "Chains".

Furthermore, in all models without chains, said models have been scaled to a real-size scale. Thus, all this process gets more realistic for the client who order to design, self-align and/or simulate the model.

Finally, it is time to self-align the mechanism, as well as to enter motion in joints for the simulation. It is recommended to generate video files of the simulation, and photos of all assemblies and parts.

Next, the process followed with LTm_8862-1 using COSMOSMotion is described.

To start using COSMOSMotion with that model, this file is copy-pasted and named with …-motion. For instance, vLTm_8862-1_2015-motion. The fixed part is the chassis of the model. The process of creating a mechanism with all its constraints can be founded in the 'Help' tab of COSMOSMotion on the top of the program. When moving parts from 'Assembly components' to 'Moving parts' some constraints are generated. It is needed to always have the 'Simulation panel' opened, because all these constraints must be removed.

Once the whole model is self-aligned, a screenshot to the full tree of constraints is taken and '…-motion' file is copied-pasted as: vLTm_8862-1_2015-motion-base. In this document, the fixed part is going to be a new part called 000000 which is the floor/base. To do so, some steps are recommended. First, the base is moved to "ground parts" and the chassis to "moving parts". By doing so, some new constraints appear. Now, to remove unwanted new constraints the screenshot file needs to be opened. So, now, all the constraints which are not in the screenshot must be removed. Second, a contact constraint is created between the base and each wheel. Notice that the more closed the base and wheels are (not tangent nor with interference), the less impact and bounces will occur. By clicking right on 'Motion Model' > 'System defaults' > 'World', the vector and value of gravity vector can be changed. Finally, simulation can take place.

8. DESIGNED AND SIMULATED MODELS

The content of this section can be found in Document 3 > Section 1 "1. DESIGNED AND SIMULATED MODELS". Here, all said in this Document 1 can be understood in practical cases.

9. CONCLUSIONS

First it is needed to highlight that, as it has already been said when introducing Doc #1 > Section 8 “DESIGNED AND SIMULATED MODELS”, “to easily show how either design, self-aligning and simulation of each model has been made, big size images appear. If two images are not possible to share page, they have been put in different pages but aligned to the middle of the page. The results of CAD and CAE programs must be displayed in a 2D format such as it is a DIN A4 page to be exported to pdf. **This is the reason why the document exceeds the 80 maximum pages of the Thesis’ Bases”.**

Apart from that, **document 2 (an excel with the formulas), *.avi and *.mp4 files of the simulations, extra images of some joints of a real machine and more can be found in the same link as indicated in previous section: https://drive.google.com/drive/folders/1Okg5BzO2PfgPe7qQNgQ-cGTj4MY-BkMS?usp=sharing**. Once said this, it is time to expound on conclusions after finishing the Thesis.

Without any doubt, this bachelor’s degree In Industrial Engineering has been a source of knowledge and learning. During all this time at the University, to acquire taught knowledge is not the only important thing to do. If there is no motivation and an active mind to realize about things and to understand how the systems work, knowledge, tools, experiences, and learnings just rot in oblivion. Thus, from the very first time this Thesis was started, all these capabilities have been satisfactorily tested. In addition, during the development of this Thesis, much more information has been acquired. In this Thesis, two topics have been worked.

First topic is “Mechanical Engineering and Computational Engineering”. Some developed actions are: design, calculus, test, and simulation of mechanical problems; to research mechanical solutions; to understand how some very common mechanism works, like the 4-bar or the 3-bar, as well as what lays the foundation of mechanisms’ motion and utility, like industrial vehicles or how to use Geometry and Mechanics to make a useful machine; to explain what are actually joints and the rest constraints; how kinematic constraints can be implemented in the physical world and in the virtual world, by using the Table of Reshetov and CAD and CAE programs; …

Second topic is “Projects”. Some developed actions are: to research new information; to cooperate with other people to move forward a Project; to solve problems by using the my own initiative; to organize and to filer information; to plan how time, effort, money and resources are going to be distributed and used, according to time line and necessities; to balance personal and professional time; to use graphic design to introduce a Project; …

Although LEGO® Technic Models (LTM) have been used in this Thesis, it is just a very practical tool to learn, explain and practice how most of mechanisms work, how kinematic constraints (like bearings, according to the “Table of Reshetov”) must be selected to create self-aligned machines, … All this can be extrapolated to any machine or mechanism. This way, more knowledge has been acquired from the Thesis’ topic:

- How internally computational aided design, CAD, and computational aided engineering, CAE, programs work. How to search information in an industrial environment. A professional point of view of these programs, like SolidWorks® 2007, COSMOSMotion 2007, SolidWorks® 2020 and RecurDyn.

- How little different options can make a huge difference in results. For example: some options can be activated or disactivated in a CAE program to generate different behaviors of the same machine even with the same geometry; or how a machine can be absolutely worthless by changing just one joint.
- A great improvement in spatial vision.
- To develop a methodology and to implement a system to improve it while following it. This is a basic concept in the professional environment, where all process and actions must be perfectly detailed.
- During the development of the Thesis, two important conclusions has been commented: (1) the importance of self-aligning a mechanism; (2) any machine can become a self-aligned mechanism if it is properly design and built. Only in two mechanisms, to get zero redundant constraints (NRC = 0) has not been possible, but it has been said why and what to do to solve it. Both can be re-designed to become self-aligned.
- Finally, the value of self-aligning mechanisms and machines has also been made aware. They largely eliminate programmed obsolescence. Thus, to do this is a very useful step forward in environmental sustainability in the machine design environment.

All this time and effort have been worth it.

All objectives of the Thesis written in Document 1 > Section 1 have been accomplished.

10. BIBLIOGRAPHY

1. **Document 1 and 2 of this Thesis, *.avi and *.mp4 files of the simulations, and more information and files.** **https://drive.google.com/drive/folders/1Okg5BzO2PfgPe7qQNgQ-cGTj4MY-BkMS?usp=sharing.**

2. **About mechanisms:**
   Information about all the theory of this Thesis and about LTM:
   2.1. Theory:
      2.1.1. Website of the tutor of the Thesis: **http://www.upv.es/vltmodels/index.html**
      2.1.2. Reshetov, L. (1982). In Mir Publishers (Ed.), “Self-aligning mechanisms”
   2.2. About LEGO® and LEGO® TECHNIC MODELS (LTM):
      2.2.1. Web of “LEGO®”: https://www.lego.com/es-es/
      2.2.2. Web of LTM: http://www.lego.com/en-us/technic
      2.2.3. http://www.isogawastudio.co.jp/
      2.2.4. Information about any model of LEGO®: https://brickset.com/
      2.2.5. “BrickLink”. It is like “the Ebay of Lego®”. Useful to search all pieces of a model, its color and reference, …: https://www.bricklink.com/v2/main.page
      2.2.6. A great blog about Lego®: www.technicbricks.com
      2.2.7. https://www.brickeconomy.com/set/8862-1/lego-technic-backhoe-grader
      2.2.8. http://www.technicopedia.com/index.html
   2.3. Design, self-aligning and simulation of some other models:
      2.3.1. “Virtual Technic Models”, a web site of the tutor of the Thesis. https://www.virtualtechmodels.com/
      2.3.2. Jose Oliver’s Grabcad public account. https://grabcad.com/jose.oliver/models
   2.4. Channel of Jose Luis Oliver Herrero in “media.upv.es”, a platform from the UPV. https://media.upv.es/#/portal/user/79754847-ff21-42ba-a005-262b6f8528ca/videos

3. **Books and libraries:**
   3.1. “Digital mechanism and gear library” (DMG Lib). A huge virtual library about mechanisms. https://www.dmg-lib.org/dmglib/main/portal.jsp?mainNaviState=aktuell.start
   3.2. “Rapid Design through Virtual and Physical Prototyping. Introduction to mechanisms”, Yi Zhang, Susan Finger and Stephannie Behrens. Carnegie Mellon University. https://www.cs.cmu.edu/~rapidproto/mechanisms/tablecontents.html
   3.3. “Engineering Books PDF”. https://engineeringbookspdf.com/
   3.4. “Access engineering library”. https://www.accessengineeringlibrary.com/
   3.5. David H Myszka; “Machines and mechanisms. Applied kinematic analysis”:
      3.5.1. https://fliphtml5.com/lnym/hqkw/basic/151-200

3.5.2. https://scholar.google.es/scholar?q=%2BMachines+and+mechanisms.+Applied+kinematic+analysis.+David+H+Myszka&hl=en&as_sdt=0&as_vis=1&oi=scholart

3.6. Ferdinand Freudenstein, Ph.D; George N. Sandor, Eng.Sc.D., P.E. "Mechanical design handbook. Chapter 3: Kinematics of mechanisms".

3.7. Artobolevsky, I. I. (1976). "Mecanismo en la técnica moderna. Tomo I"

4. About degrees of freedom:

5. Wikipedia.

5.1.1. https://en.wikipedia.org/wiki/Degrees_of_freedom_(mechanics)

6. "Structural synthesis of robot manipulators by using screw with variable pitch", Rasim Alizade. Azerbaijan Technical University.

6.1.1. http://web.iyte.edu.tr/~gokhankiper/ISMMS/Alizade.pdf

7. About CAE programs:

7.1. About "RecurDyn" in general:

7.1.1. Main page. https://support.functionbay.com/en

7.1.2. FAQs > "How can I remove redundant constraints?" https://support.functionbay.com/en/faq/single/105/can-remove-redundant-constraints

7.1.3. Basics of the contact algorithm:

7.1.3.1. https://support.functionbay.com/en/technical-tip/single/224

7.1.3.2. https://www.youtube.com/watch?v=fXRw3ebJYSQ&ab_channel=RecurDyn

7.2. About the toolkit "Track_LM" (chains):

7.2.1. https://functionbay.com/documentation/onlinehelp/default.htm#!Documents/sprocket1.htm

7.2.2. https://www.youtube.com/watch?v=etEZEP4Lkk0

7.3. "EnginSoft", a company in the field of Computer Aided Engineering (CAE). https://enginsoftusa.com/

7.4. "Linkage". A program to design planar mechanisms. https://blog.rectorsquid.com/

7.5. "Creo Parametric" Online Help:

7.5.1. About the kinematic analysis. http://support.ptc.com/help/creo/creo_pma/usascii/index.html#page/simulate/mech_des/analysis/AboutKinematicAnalysis.html#wwconnect_header

7.5.2. About predefined constraints sets. http://support.ptc.com/help/creo/creo_pma/usascii/index.html#page/simulate%2Fmech_des%2Fconnections%2FAboutPredefinedConstraintSets.html%23

7.5.3. About degrees of freedom of the predefined constraints sets. http://support.ptc.com/help/creo/creo_pma/usascii/index.html#page/simulate%2Fmech_des%2Fconnections%2Fdegs_freedom.html%23

7.5.4. About redundancies.

http://support.ptc.com/help/creo/creo_pma/usascii/index.html#page/simulate%2Fmech_des%2Fconnections%2Fdof_redundancies.html%23

7.6. An example of how to calculate degrees of freedom and redundancies.

7.6.1. http://support.ptc.com/help/creo/creo_pma/usascii/index.html#page/simulate%2Fmech_des%2Fconnections%2Fcalculating_dof_redund.html%23

8. Links of interest:

8.1. LEGO Antikythera Mechanism (Ancient Greek Computer).
https://www.youtube.com/watch?v=t1I1kdW3wgE

8.2. Prof. Jonathan Hopkins at UCLA:

8.2.1. Introduction to the method “Freedom And Constraint Topologies” (FACT).
https://www.youtube.com/watch?v=BnWRI7oTBeQ

8.2.2. A way to design compliant mechanisms according to needed degrees of freedom.
https://www.nature.com/articles/s41467-018-08049-1

8.2.3. Range of net force = 0 in mechanisms.
https://www.youtube.com/watch?v=uvPSsUkLypY&ab_channel=TheFACTsofMechanicalDesign

8.3. Examples of mechanisms created by “thang010146”.
https://www.youtube.com/channel/UCli_RJkGWfZvw4IlDLHNCQg

8.4. Golovin, A., & Tarabarin, V. “Russian models from the mechanisms collection of Bauman University”

8.5. Flexure joints for large range of motion by Precision Engineering lab at the University of Twente:

8.5.1. https://www.youtube.com/watch?v=4fFH2RHpcTY&ab_channel=PrecisionEngineering%2CUniversityofTwente

8.5.2. https://www.youtube.com/watch?v=i8Ad-qi9q7A&ab_channel=PrecisionEngineering%2CUniversityofTwente

8.6. “Pic Design”, a premium quality precision mechanical components.
https://www.pic-design.com/

8.7. A detailed description of some gears. AUSV 2520.
https://www.youtube.com/watch?v=5piYEX-jRt4&ab_channel=WeberAuto

8.8. More about LEGO®:

8.8.1. https://jkbrickworks.com/da-vinci-flying-machine/

8.8.2. https://www.nico71.fr/category/farmequipment/

8.9. Some fundaments of mechanisms and joints can be observed in real mechanisms:

8.9.1. https://www.youtube.com/watch?v=rcP7ZvzyZwM&ab_channel=JerryRigEverything

8.9.2. https://www.youtube.com/watch?v=b0DKNIQFuzg&ab_channel=DanRose

8.9.3. https://www.youtube.com/watch?v=aKkLBK22nZc&ab_channel=AmazingTechnology

8.9.4. https://www.youtube.com/watch?v=den7s6CujxA&ab_channel=WojciechZaczek

8.10. “Can LIGHT pick stuff up and assemble it?”. To create new atomic arrangements.
https://www.youtube.com/watch?v=2PD48xgHvjs

BACHELOR’S DEGREE IN INDUSTRIAL ENGINEERING. BACHELOR’S THESIS

DESIGN AND REAL-SCALE COMPUTATIONAL SIMULATION OF INDUSTRIAL VEHICLES BASED ON MODELS LTM 42043-2 AND LTM 42028-2

DOCUMENT 2.- BUDGET

AUTHOR:	MIGUEL SEGARRA MARTÍNEZ
SUPERVISOR:	JOSE LUIS OLIVER HERRERO

Academic year: 2021-22

INDEX 2

LIST OF FIGURES 2

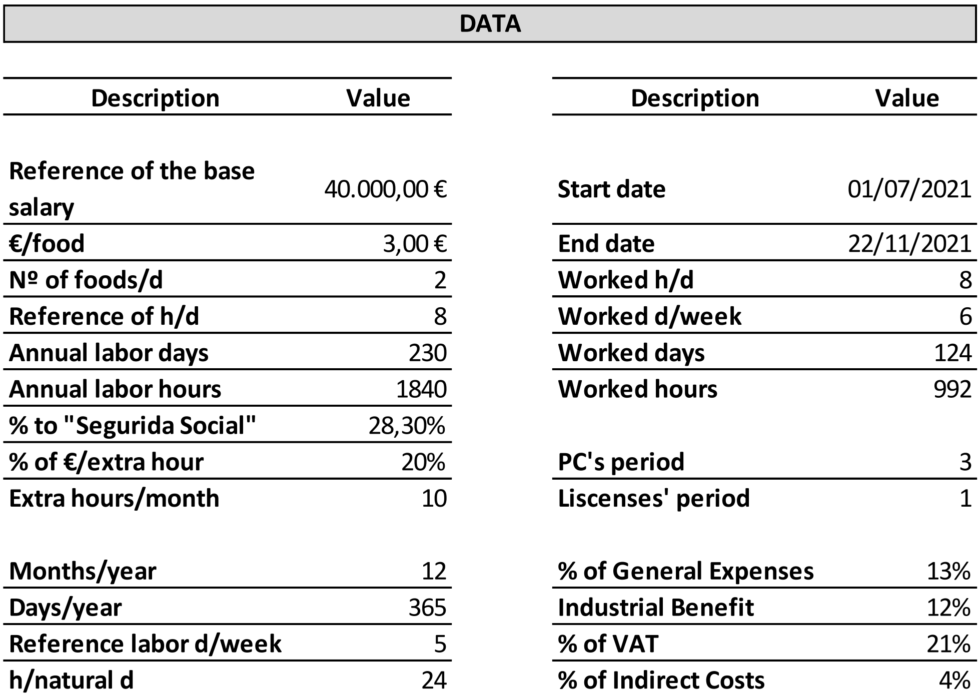
1. THE TOTAL BUDGET

In the labor force, only the hours invested by senior engineer will be taken into account, who will be responsible for (1) the creation of the components needed to assembly all models, (2) for the monitoring of a correct assembly and a self-aligning process, and (3) for the analysis and validation of the final documents generated by the author of the Thesis.

According to the Spanish ministerial order "*Orden ESS/55/2018, de 26 de enero, por la que se desarrollan las normas legales de cotización a la Seguridad Social, desempleo, protección por cese de actividad, Fondo de Garantía Salarial y formación profesional para el ejercicio 2018*" published in the Spanish Official State Gazette ("*Boletín Oficial del Estado",* BOE) on January 29th, 2018, a ratio f contribution of 28.3% to the Social Security in terms of common contingencies is established. A total of 46 work weeks per year will be considered, equivalent to the corresponding 230 working days of 8 hours, and 10 extra hours per month with an additional price of 20%.

Taking as reference the provisions of the "*XVIII Convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos*" published in the BOE on January 18th, 2017, the following value will be considered: 3€ per day as food diet.

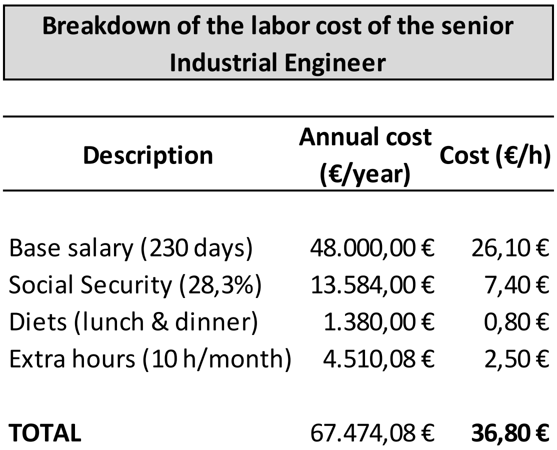
The document used to calculate the budget can be found in the same link that simulations' videos: https://drive.google.com/drive/folders/1Okg5BzO2PfgPe7qQNgQ-cGTj4MY-BkMS?usp=sharing



DATA

Description	Value	Description	Value
Reference of the base salary	40.000,00 €	**Start date**	01/07/2021
€/food	3,00 €	**End date**	22/11/2021
Nº of foods/d	2	**Worked h/d**	8
Reference of h/d	8	**Worked d/week**	6
Annual labor days	230	**Worked days**	124
Annual labor hours	1840	**Worked hours**	992
% to "Segurida Social"	28,30%		
% of €/extra hour	20%	**PC's period**	3
Extra hours/month	10	**Liscenses' period**	1
Months/year	12	**% of General Expenses**	13%
Days/year	365	**Industrial Benefit**	12%
Reference labor d/week	5	**% of VAT**	21%
h/natural d	24	**% of Indirect Costs**	4%

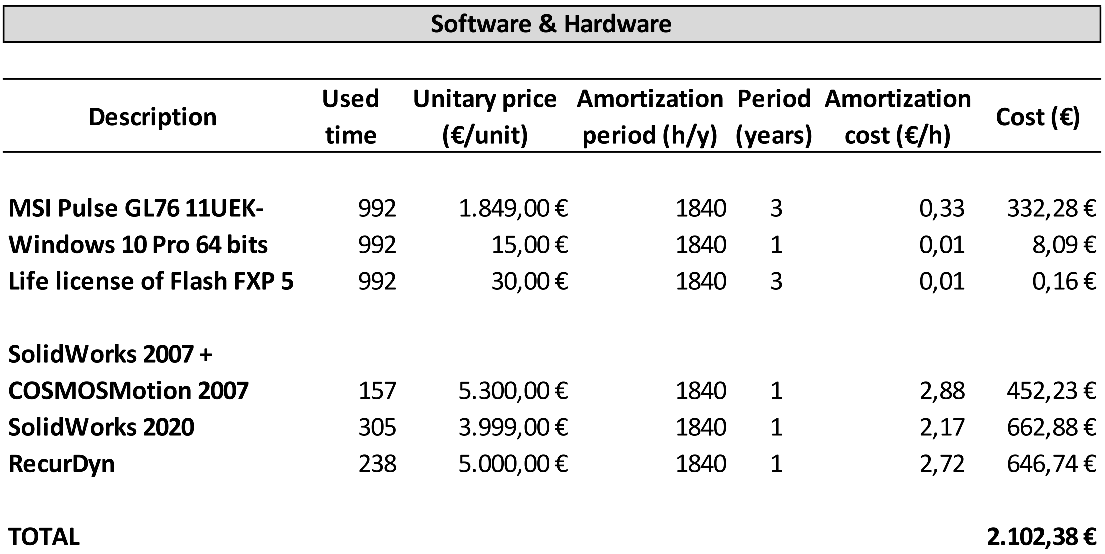
Figure 66. Data and values used to calculate the budget.



Breakdown of the labor cost of the senior Industrial Engineer

Description	Annual cost (€/year)	Cost (€/h)
Base salary (230 days)	48.000,00 €	26,10 €
Social Security (28,3%)	13.584,00 €	7,40 €
Diets (lunch & dinner)	1.380,00 €	0,80 €
Extra hours (10 h/month)	4.510,08 €	2,50 €
TOTAL	67.474,08 €	**36,80 €**

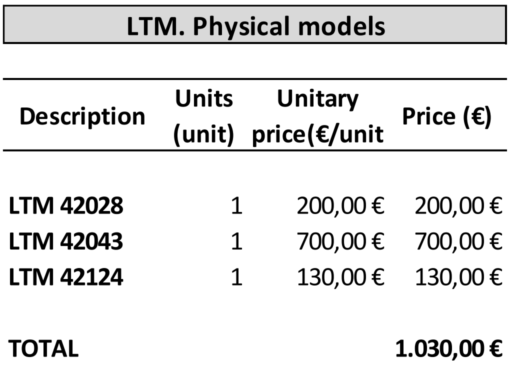
Figure 67. Breakdown of the labor cost of the senior Industrial Engineering.



Software & Hardware

Description	Used time	Unitary price (€/unit)	Amortization period (h/y)	Period (years)	Amortization cost (€/h)	Cost (€)
MSI Pulse GL76 11UEK-	992	1.849,00 €	1840	3	0,33	332,28 €
Windows 10 Pro 64 bits	992	15,00 €	1840	1	0,01	8,09 €
Life license of Flash FXP 5	992	30,00 €	1840	3	0,01	0,16 €
SolidWorks 2007 + COSMOSMotion 2007	157	5.300,00 €	1840	1	2,88	452,23 €
SolidWorks 2020	305	3.999,00 €	1840	1	2,17	662,88 €
RecurDyn	238	5.000,00 €	1840	1	2,72	646,74 €
TOTAL						**2.102,38 €**

Figure 68. The sum of "Software & Hardware" costs.



LTM. Physical models

Description	Units (unit)	Unitary price(€/unit	Price (€)
LTM 42028	1	200,00 €	200,00 €
LTM 42043	1	700,00 €	700,00 €
LTM 42124	1	130,00 €	130,00 €
TOTAL			**1.030,00 €**

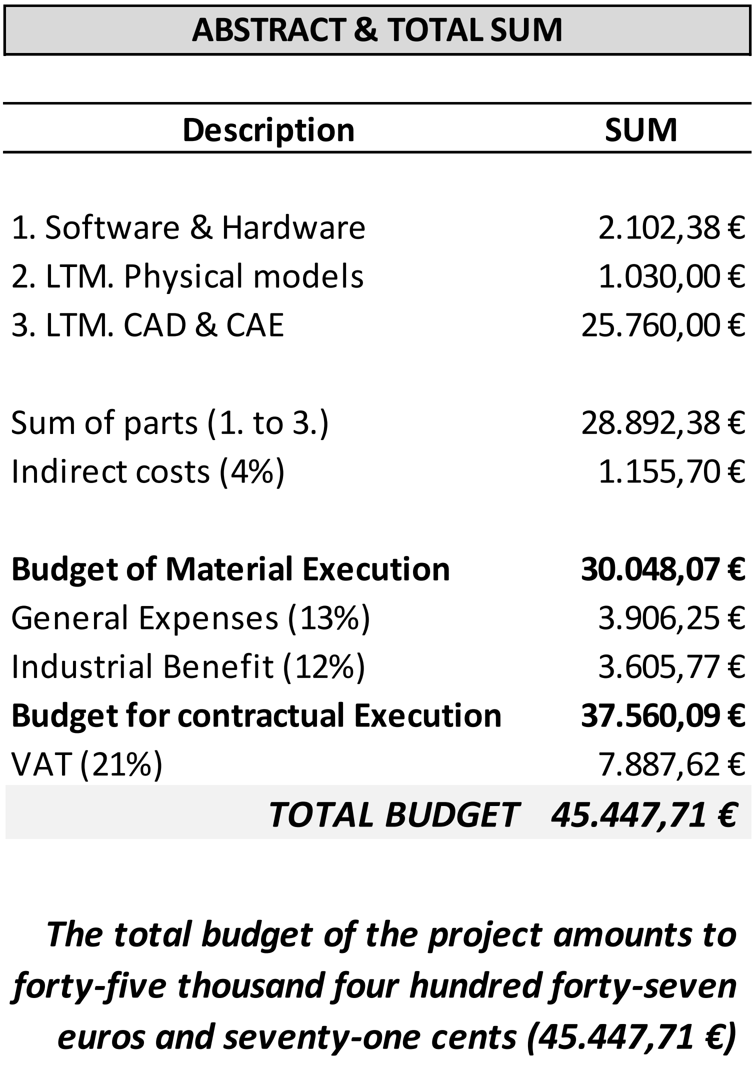
Figure 69. The sum of "LTM. Physical models" costs. Once they are physically assembled, they are virtualized.

LTM. CAD & CAE

Description	Working time (h)	Unitary price (€/h)	Price (€)
LTM 42028			
Assembly in SW 2020	45	36,80 €	1.656,00 €
Self-aligning in RecurDyn	25	36,80 €	920,00 €
Simulating in RecurDyn	20	36,80 €	736,00 €
	90		**3.312,00 €**
LTM 42043			
Assembly in SW 2020	260	36,80 €	9.568,00 €
Self-aligning in RecurDyn	70	36,80 €	2.576,00 €
Simulating in RecurDyn	26	36,80 €	956,80 €
	356		**13.100,80 €**
LTM 8862			
Assembly in SW 2007	27	36,80 €	993,60 €
Self-aligning in COSMOS	24	36,80 €	883,20 €
Simulating in COSMOS	13	36,80 €	478,40 €
	64		**2.355,20 €**
LTM 8047-1			
Assembly in CAD	-	36,80 €	0,00 €
Self-aligning in RecurDyn	29	36,80 €	1.067,20 €
Simulating in RecurDyn	17	36,80 €	625,60 €
	46		**1.692,80 €**
LTM 8294-1			
Assembly in CAD	-	36,80 €	0,00 €
Self-aligning in RecurDyn	31	36,80 €	1.140,80 €
Simulating in RecurDyn	20	36,80 €	736,00 €
	51		**1.876,80 €**
LTM 42124-1			
Assembly in SW 2007	38	36,80 €	1.398,40 €
Self-aligning in COSMOS	31	36,80 €	1.140,80 €
Simulating in COSMOS	24	36,80 €	883,20 €
	93		**3.422,40 €**
TOTAL	**700**		**25.760,00 €**

Figure 70. The sum of "LTM. CAD & CAE" costs.

The rest of the time has been used to redact the Thesis, to search for the needed and more knowledge, to correct mistakes and evaluate go-no-go points, to think about an algorithm to self-align any mechanism, … Time has not been wasted. Maybe, too much time has been spent in looking for perfection.



ABSTRACT & TOTAL SUM

Description	SUM
1. Software & Hardware	2.102,38 €
2. LTM. Physical models	1.030,00 €
3. LTM. CAD & CAE	25.760,00 €
Sum of parts (1. to 3.)	28.892,38 €
Indirect costs (4%)	1.155,70 €
Budget of Material Execution	**30.048,07 €**
General Expenses (13%)	3.906,25 €
Industrial Benefit (12%)	3.605,77 €
Budget for contractual Execution	**37.560,09 €**
VAT (21%)	7.887,62 €
TOTAL BUDGET	***45.447,71 €***

The total budget of the project amounts to forty-five thousand four hundred forty-seven euros and seventy-one cents (45.447,71 €)

Figure 71. The abstract & the total budget of the Project.

BACHELOR'S DEGREE IN INDUSTRIAL ENGINEERING. BACHELOR'S THESIS

DESIGN AND REAL-SCALE COMPUTATIONAL SIMULATION OF INDUSTRIAL VEHICLES BASED ON MODELS LTM 42043-2 AND LTM 42028-2

DOCUMENT 3.- ANNEX

AUTHOR: MIGUEL SEGARRA MARTÍNEZ
SUPERVISOR: JOSE LUIS OLIVER HERRERO

Academic year: 2021-22

INDEX 3

LIST OF FIGURES 3

1. DESIGNED AND SIMULATED MODELS

About this section. In this section, it is going to be shown how all models have been designed, self-aligned and simulated. Next, there are some aspects:

- Only the first two machines are the ones in the title of the Thesis. So, only the assemblies and parts in these two models will appear. Other machines are shown because they have used to practice more. All the knowledge exposed in Document 1 had to be practiced for a long time in very different mechanisms.

- The physical aspects of the models introduced next are not the most important things about this Thesis, that is to say, some images of real machines which are kinematically designed in a very similar way will appear. It is very interesting to analyze real machines after understanding how the mechanisms of the Lego® models work.

- This Thesis is not supposed to be printed. And very different graphical information will be given through this section. It complements the Document 1. Thus, a lot of pages and space will be used.

- The order of titles and sub-titles is made such in a way that only needed assemblies appear. Once these pieces are introduced in a mechanism, they can be imported into the file of Recurdyn where the whole model is going to be simulated. It makes easier to self-align kinematic chain by kinematic chain. For example, the piece nº 001 of the first model is going to be introduced in the mechanism "all opened kinematic chains". In later mechanisms, it does not appear because it has already been imported.

- To easily show how either design, self-aligning and simulation of each model has been made, big size images appear. If two images are not possible to share page, they have been put in different pages but aligned to the middle of the page. The results of CAD and CAE programs have to be displayed in a 2D format such as it is a DIN A4 page to be exported to pdf.

- The fixed to the ground piece is going to be identified as "FTTGP". It is not going to appear in figure because it can be easily over-understood which piece has to be considered as FTTGP in order to do the self-aligning. It only be indicated in the caption of the first figure of the mechanism.

- All simulations are made with respect to the base. From now on, the base only may appear in the mechanisms called like "opened kinematic chains". **The *.avi and *.mp4 files of all the simulations (and more about this Thesis) can be found and downloaded from this link:** **https://drive.google.com/drive/folders/1Okg5BzO2PfgPe7qQNgQ-cGTj4MY-BkMS?usp=sharing**

1.1. MODEL #5: LTM_42028-2

This model is going to be used to explain in more detail how all the information given until this point of the document is applied in a real case.

It is a **trench digger**.

1.1.1. ASSEMBLY: LTM_42028-2

Figure 72. LTM_42028-2 in real life. Source: [this is this source](enstock3w.top/) (enstock3w.top/).

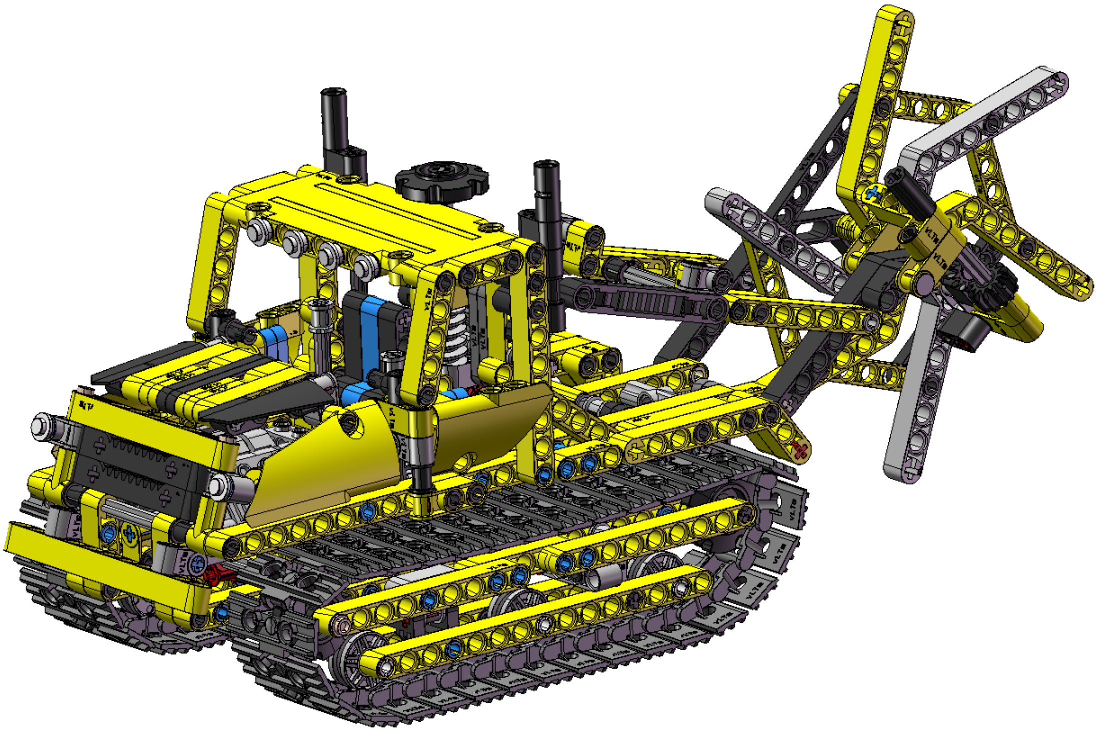


Figure 73. LTM_42028-2 in SolidWorks® 2020.

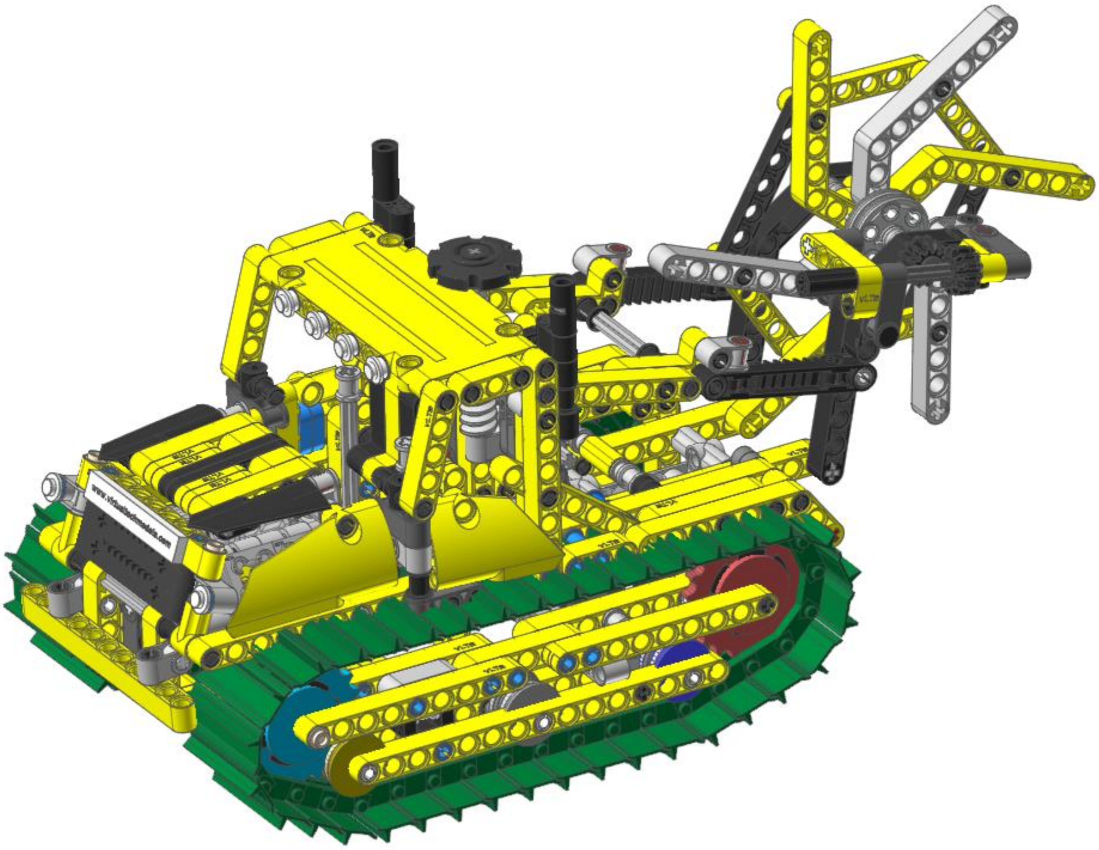


Figure 74. LTM_42028-2 after using the toolkit "Track_LM" in Recurdyn.

Figure 75. LTM_42028-2 > An industrial model in real life (#1): Marais ST2. Source: this is the source (samarais.com).

Figure 76. LTM_42028-2 > An industrial model in real life (#2): Marais GD2. Source: this is the source (samarais.com).

Figure 77. LTM_42028-2 > An industrial model in real life (#3). Source: this is the source (imgur.com/).

Figure 78. LTM_42028-2 > An industrial model in real life (#4). Source: this is the source (bossmachinery.nl).

Figure 79. LTM_42028-2 > An industrial model in real life (#5).Source: this is the source (alibaba.com).

Figure 80. LTM_42028-2 > An industrial model in real life (#6). Source: this is the source (designingbuildings.co.uk).

Figure 81. LTM_42028-2 > All assemblies in the CAD file where LTM_42028-2 is assembled using mates. Assemblies from 038 to 097 are deleted because they are the same single flange.

Figure 82. LTM_42028-2 > All the 45 parts of the assembly nº 001.

Figure 83. LTM_42028-2 > All parts of the assemblies nº 032, 022, 034 and 036.

Figure 84. LTM_42028-2 > All the components (1/2).

Figure 85. LTM_42028-2 > All the components (2/2).

1.1.2. SELF-ALIGNING MECHANISMS: VLTM_42028-2

1.1.2.1. MECHANISM “ALL OPENED KINEMATIC CHAINS”, A

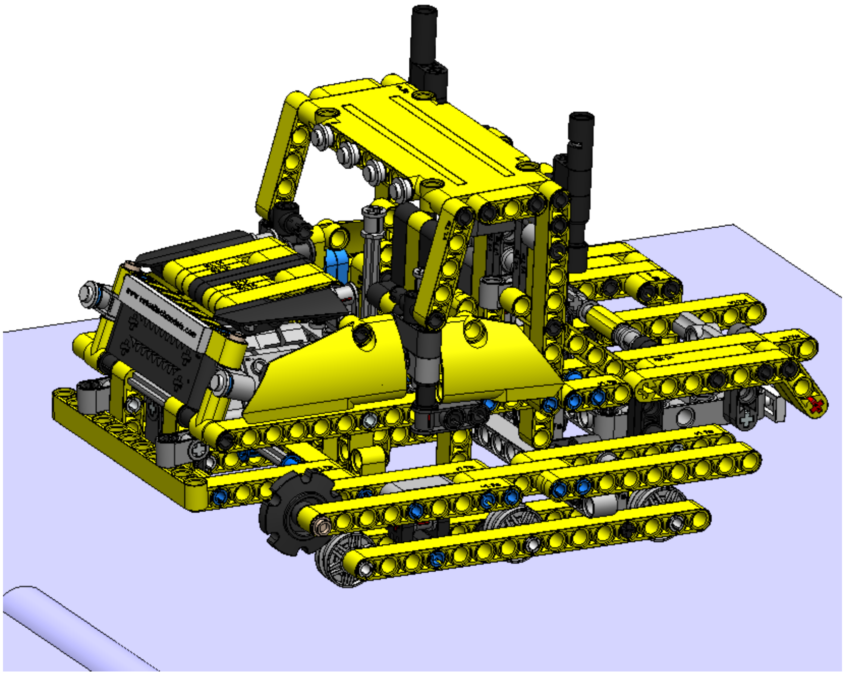


Figure 86. LTM_42028-2 > Mechanism “All opened kinematic chains” (A). These pieces are not needed to close a kinematic chain. The base is fixed with respect to the Ground; and chassis has 6 DOF as free body.

Figure 87. LTM_42028-2 > “A” > All the assemblies.

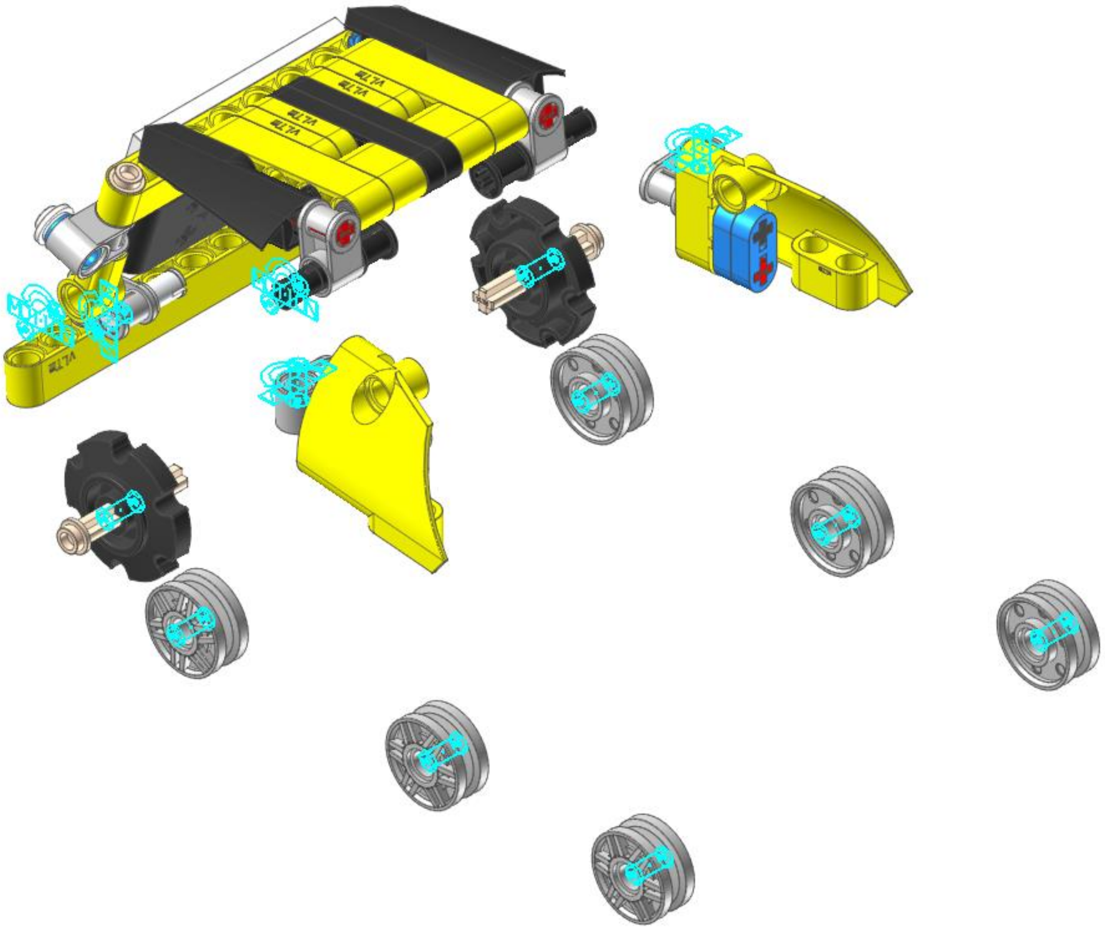


Figure 88. LTM_42028-2 > Self-aligned "A" in Recurdyn. These pieces are not needed to close a kinematic chain. 13 rotational DOF. 6 passive DOF (6 single flanges), 2 needed DOF to operate the vehicle (2 independent sprockets), and 5 auxiliar DOF (2 doors, 1 hood, 1 bumper, and 1 front). All are simulated with R joints. Also, +6 DOF because of chassis 001 as free body. FTTGP: 001

All the joints from this mechanism are defined with respect to the chassis. It could be said that, in order to self-align this mechanism, the chassis has been considered as the fixed to the ground piece. The shown pieces are moving parts with respect to the chassis. So, the chassis does not appear in this figure because it easy understandable that the chassis is one of the pieces which are defined to define a constraint. A constraint needs two pieces, so the other one is the one in the figure.

Thus, from now on, the fixed to the ground piece (FTTG) needed to self-align the considered the considered kinematic chain is not going to be shown in none of the following figures in any other model.

1.1.2.2. MECHANISM “POWER”, P

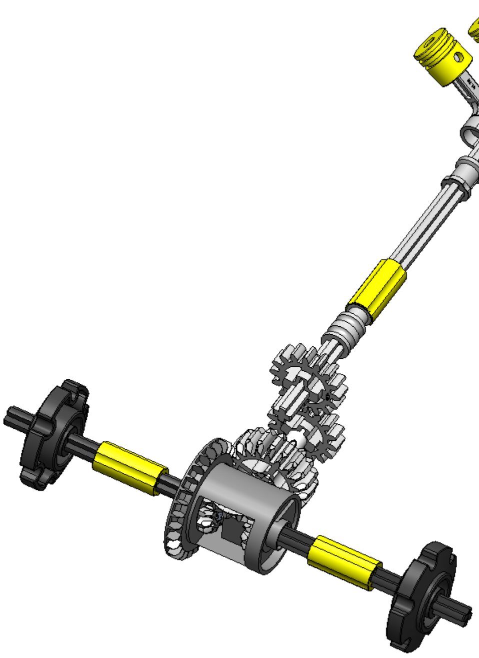


Figure 89. LTM_42028-2 > Mechanism “Power” (P) in SolidWorks® 2020. 2 rotational DOF to be controlled: 2 driving sprockets. Or 1 rotational DOF to be controlled. 1 crankshaft. FTTG: 001

In “real life”, it only would be 1 rotational DOF to be controlled: crankshaft. The power through this piece goes to sprockets, and because of the contact with the floor, sprockets would move. It also could be done in the simulation, but the vehicle might would not be very easy to manage/simulate.

Figure 90. LTM_42028-2 > “P” > All the assemblies.

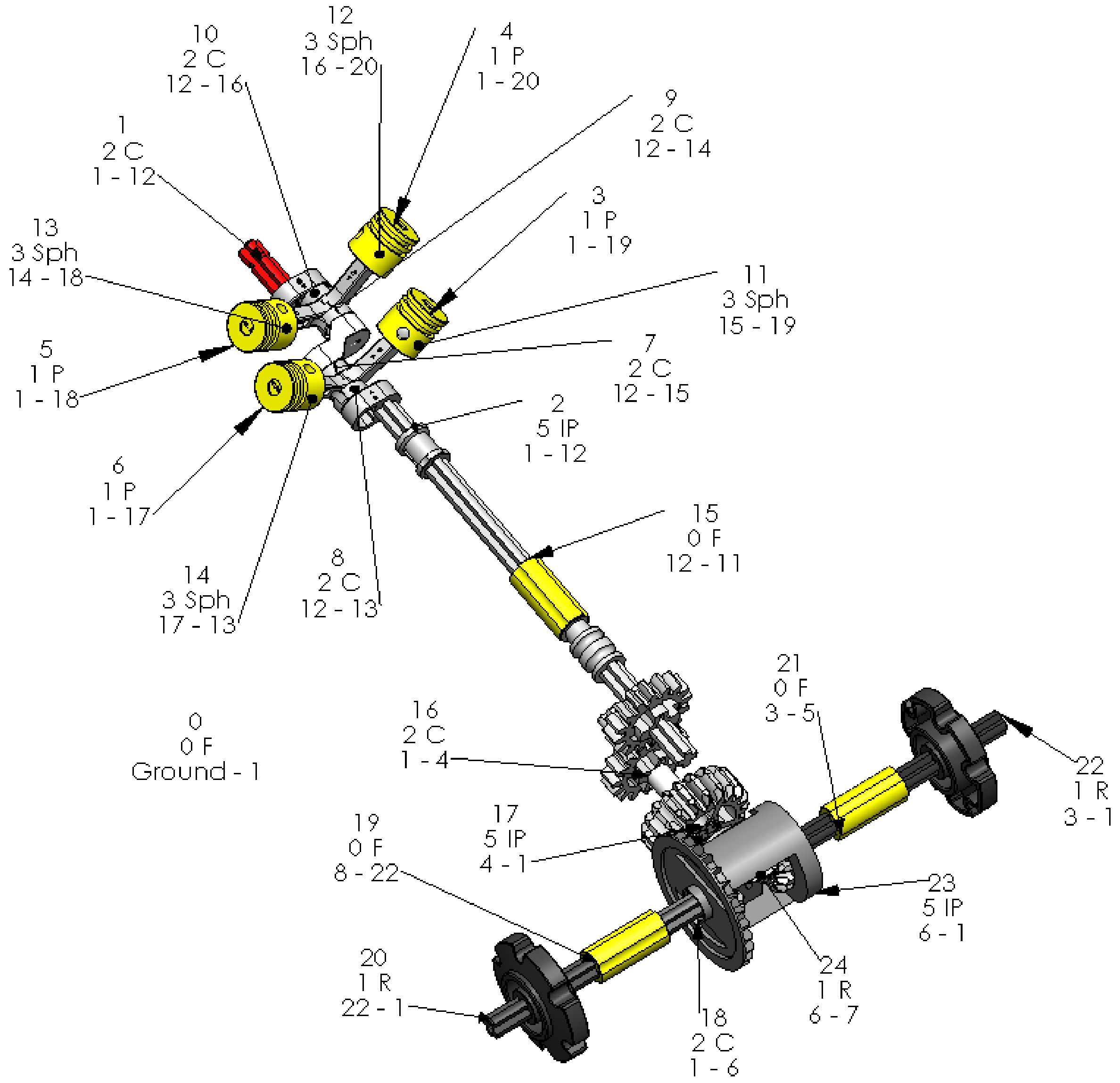
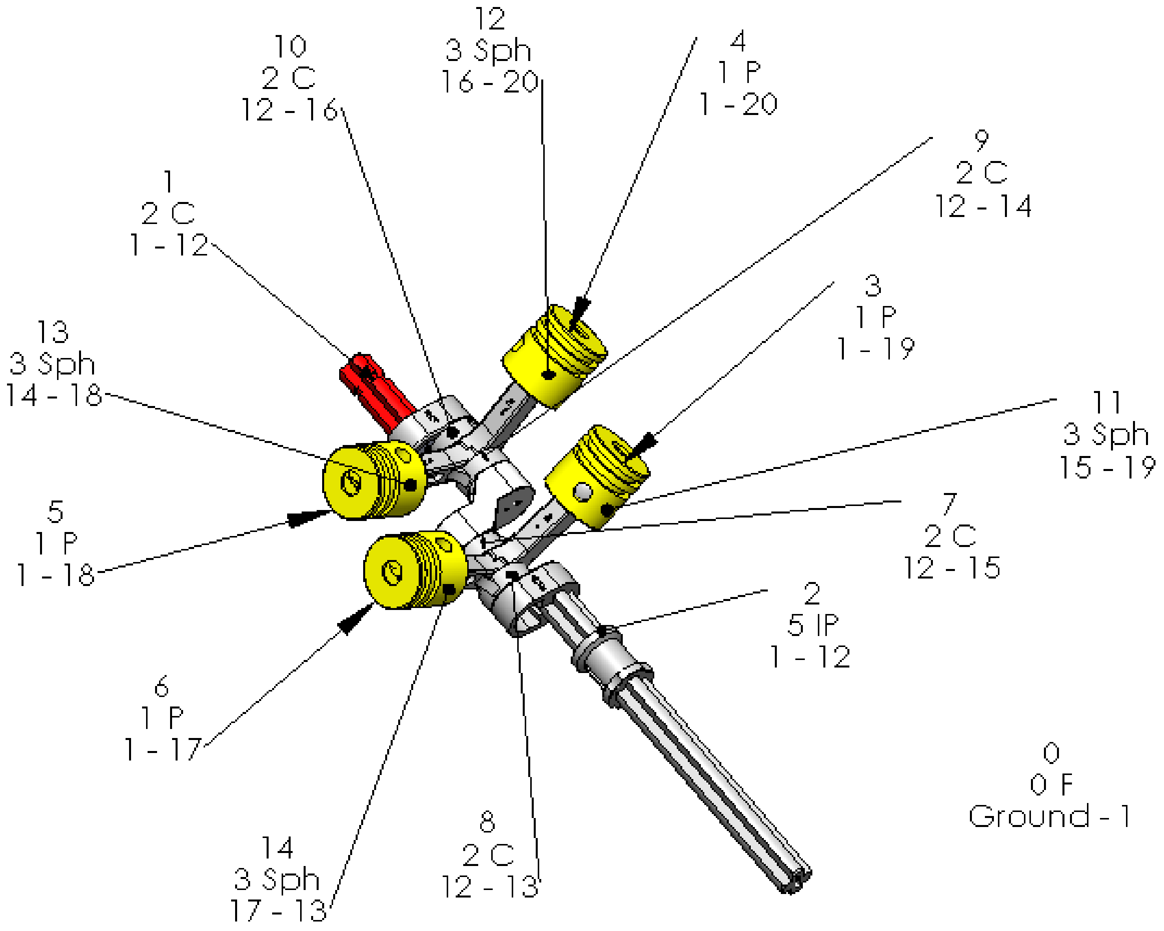


Figure 91. LTM_42028-2 > "P" > Joints in SolidWorks® 2020.



1st KC:
Nb = 10 (1 is fixed; 12, 13, ... 20)
Nco = 14 (1, 2, ..., 14) (#0 is F)
Sumat(f_i) = 0F*1 + 1P*4 + 2C*5 + 3Sph*4 + 5IP*1 = 31
Sumat(6-f_i) = 6F*1 + 5P*4 + 4C*5 + 3Sph*4 + 1IP*1 = 59

Map = 1
M3D = 6(Nb-Nco-1)+Sumat(f_i)= 1
M3D = 6*Nb - Sumat(6-f_i) = 1

So, it is needed one M in joint #1

Figure 92. LTM_42028-2 > "P" > "Kinematic diagram of the 1st Kinematic chain" in SolidWorks® 2020.

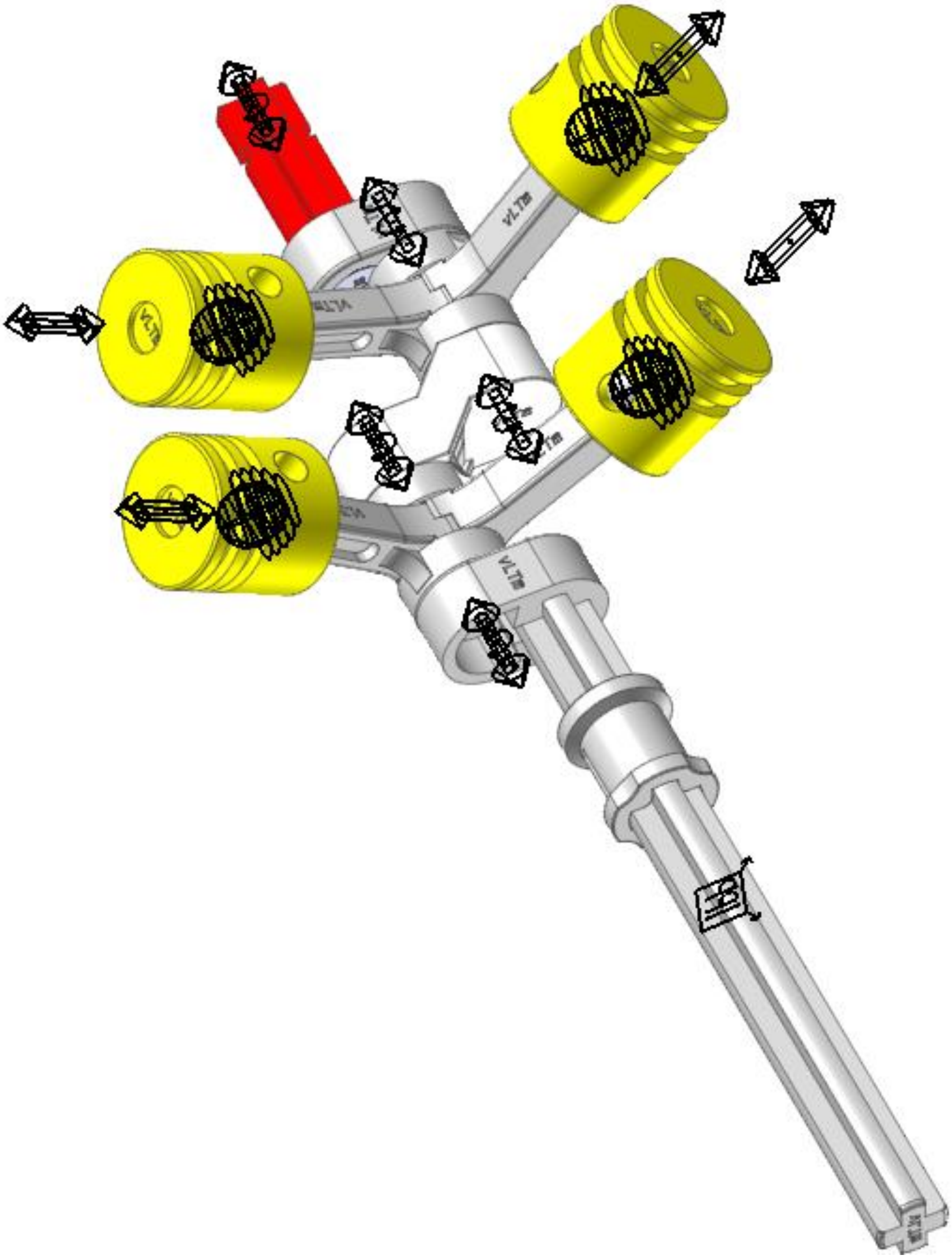
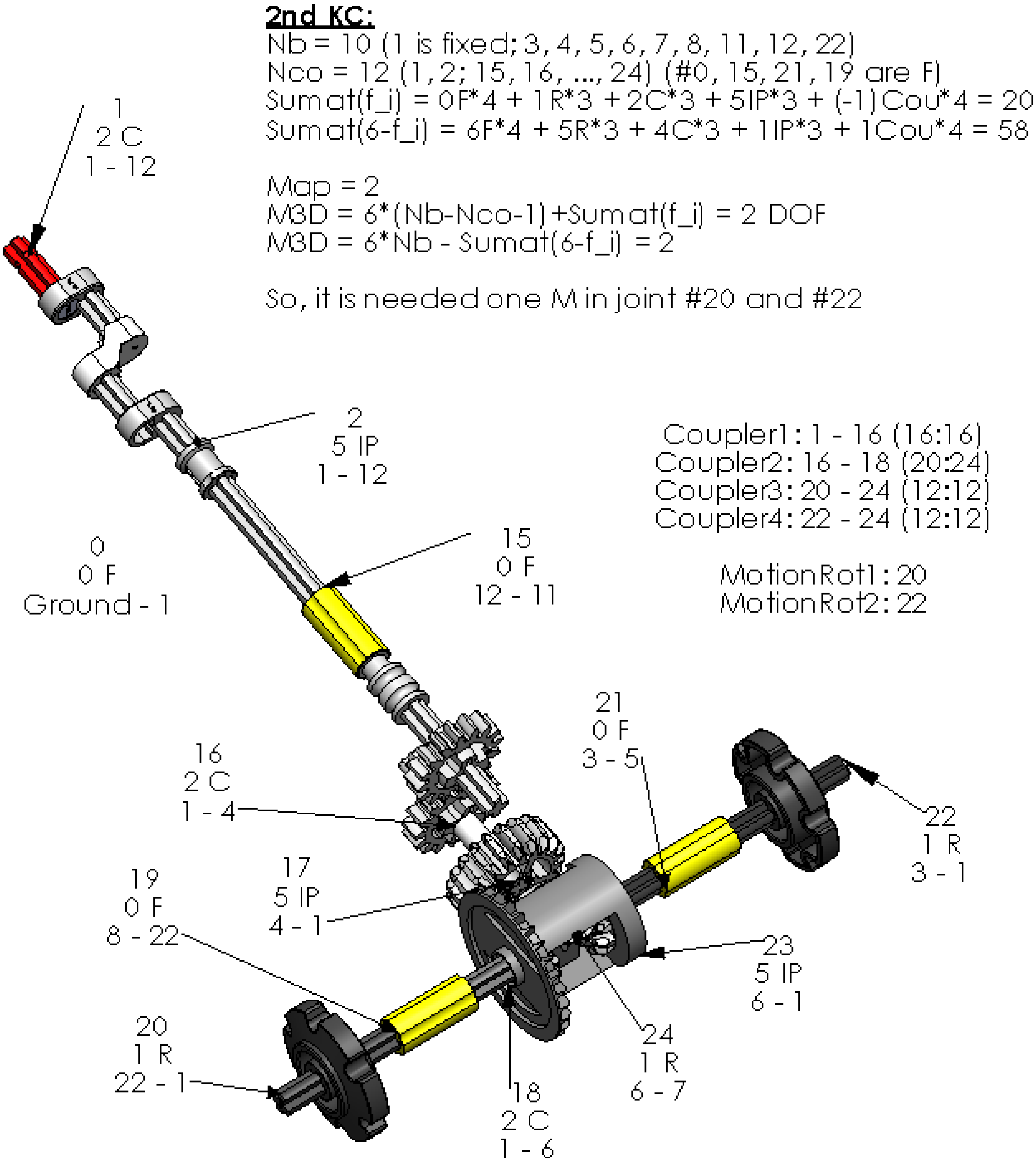


Figure 93. LTM_42028-2 > "P" > Self-aligned "Kinematic chain #1" in Recurdyn.



2nd KC:
Nb = 10 (1 is fixed; 3, 4, 5, 6, 7, 8, 11, 12, 22)
Nco = 12 (1, 2; 15, 16, ..., 24) (#0, 15, 21, 19 are F)
Sumat(f_i) = 0F*4 + 1R*3 + 2C*3 + 5IP*3 + (-1)Cou*4 = 20
Sumat(6-f_i) = 6F*4 + 5R*3 + 4C*3 + 1IP*3 + 1Cou*4 = 58

Map = 2
M3D = 6*(Nb-Nco-1)+Sumat(f_i) = 2 DOF
M3D = 6*Nb - Sumat(6-f_i) = 2

So, it is needed one M in joint #20 and #22

Coupler1: 1 - 16 (16:16)
Coupler2: 16 - 18 (20:24)
Coupler3: 20 - 24 (12:12)
Coupler4: 22 - 24 (12:12)

MotionRot1: 20
MotionRot2: 22

Figure 94. LTM_42028-2 > "P" > "Kinematic diagram of the 2nd Kinematic chain" in SolidWorks® 2020.

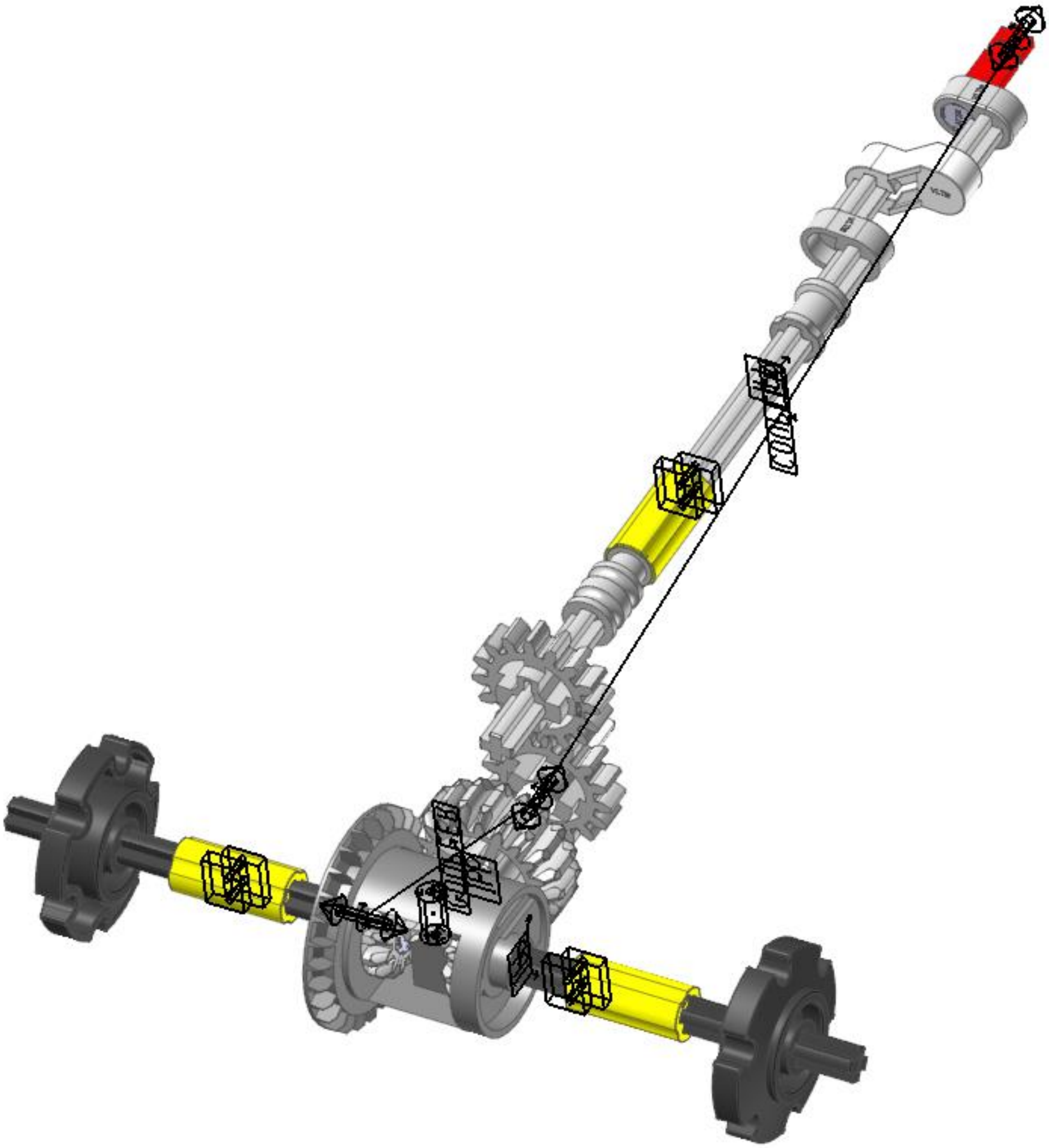


Figure 95. LTM_42028-2 > "P" > Self-aligned "Kinematic chain #2" in Recurdyn.

1.1.2.3. MECHANISM “WHEEL”, W

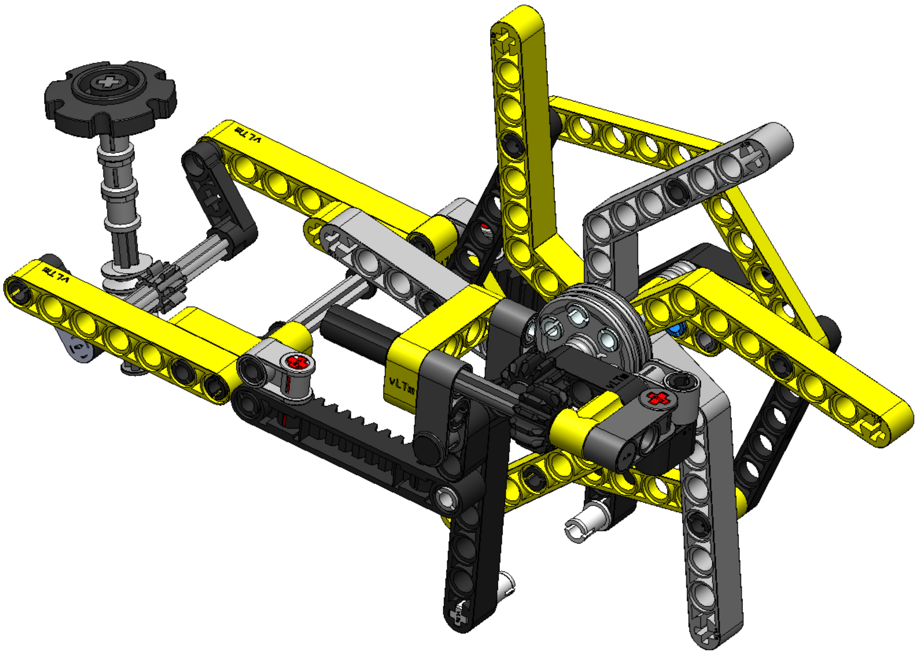


Figure 96. LTM_42028-2 > Mechanism “Wheel” (W) in SolidWorks® 2020. 2 rotational DOF to be controlled: up or down, and rotation on or off. FTTG: 001.

Figure 97. LTM_42028-2 > “W” > All the assemblies.

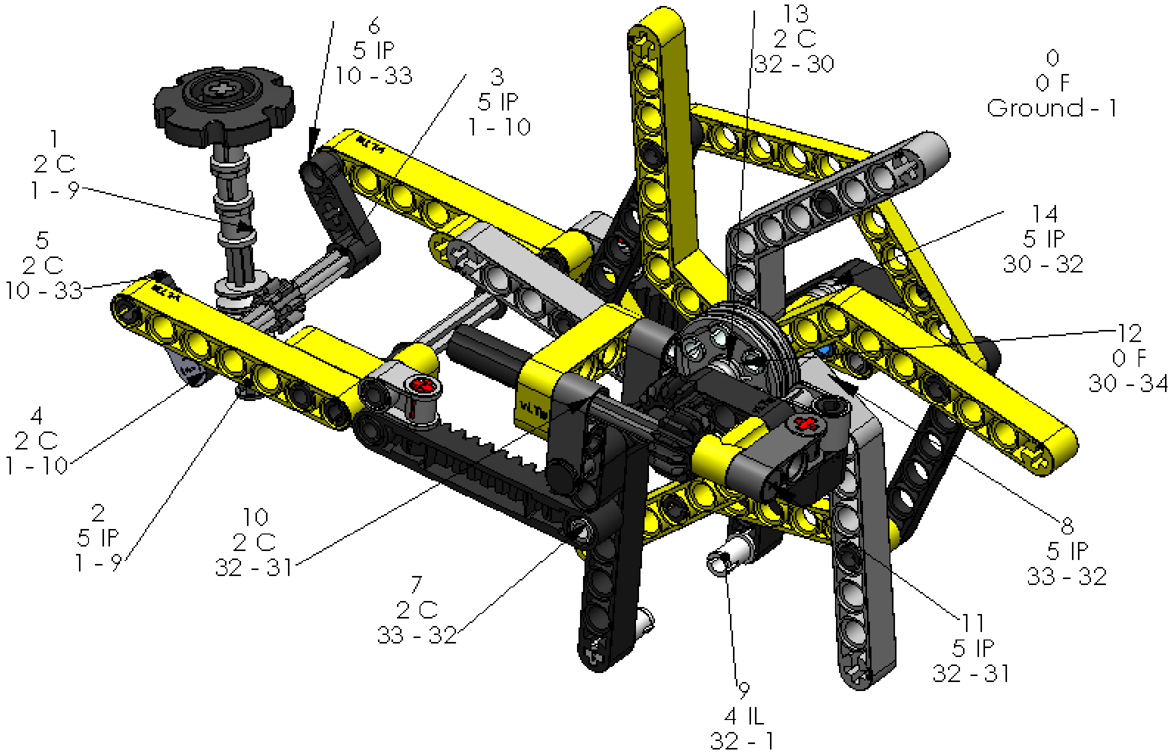
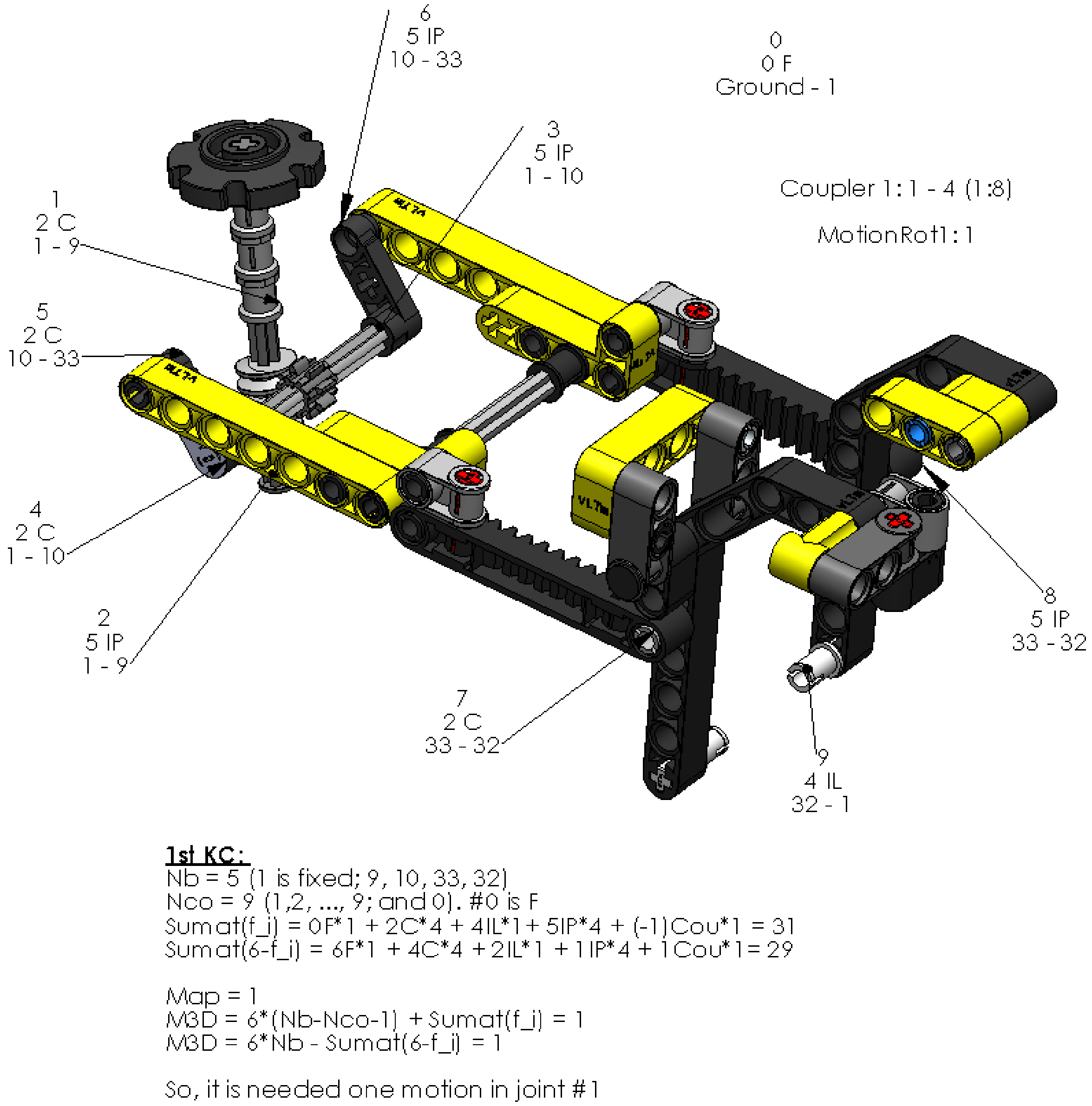


Figure 98. LTM_42028-2 > Joints of "W" in SolidWorks® 2020



1st KC:
Nb = 5 (1 is fixed; 9, 10, 33, 32)
Nco = 9 (1,2, ..., 9; and 0). #0 is F
Sumat(f_i) = 0F*1 + 2C*4 + 4IL*1+ 5IP*4 + (-1)Cou*1 = 31
Sumat(6-f_i) = 6F*1 + 4C*4 + 2IL*1 + 1IP*4 + 1Cou*1= 29

Map = 1
M3D = 6*(Nb-Nco-1) + Sumat(f_i) = 1
M3D = 6*Nb - Sumat(6-f_i) = 1

So, it is needed one motion in joint #1

Figure 99. LTM_42028-2 > "W" > "Kinematic diagram of the 1st Kinematic chain" in SolidWorks® 2020.

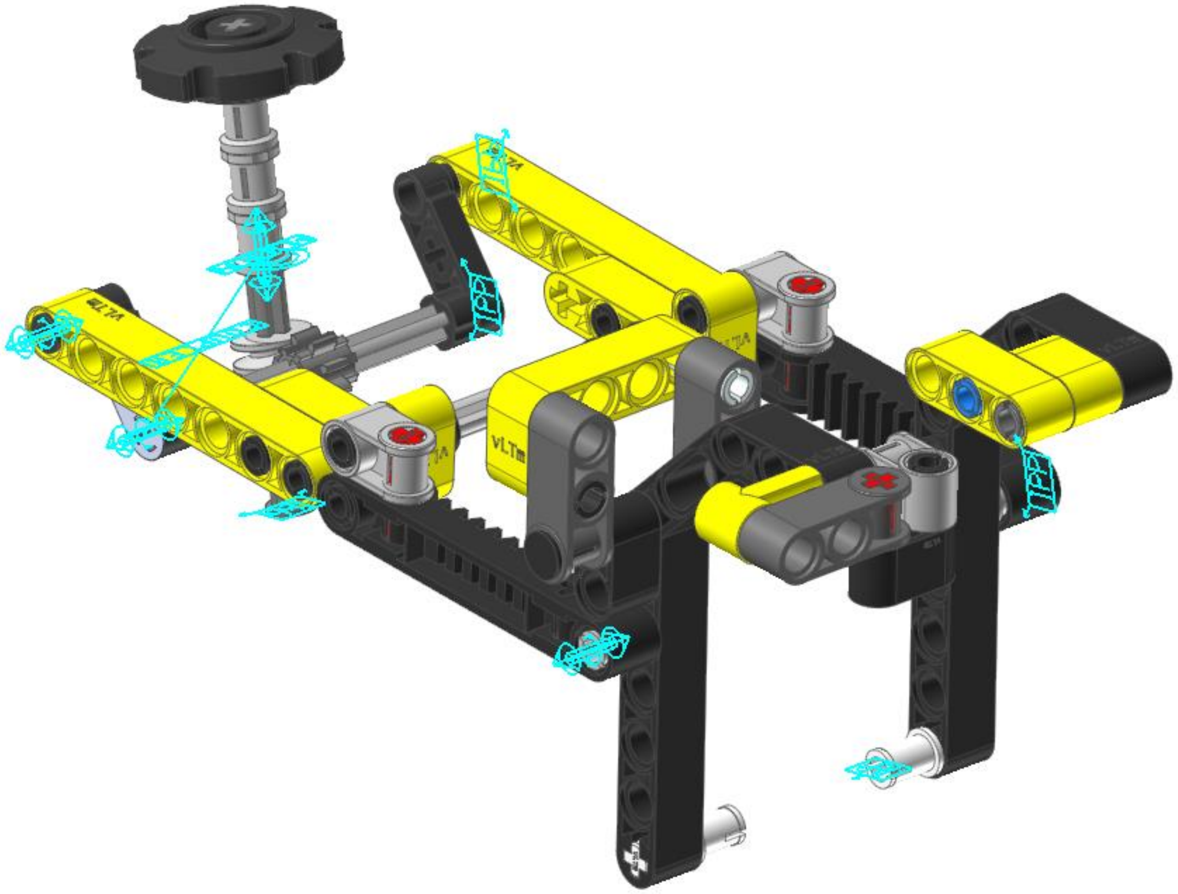
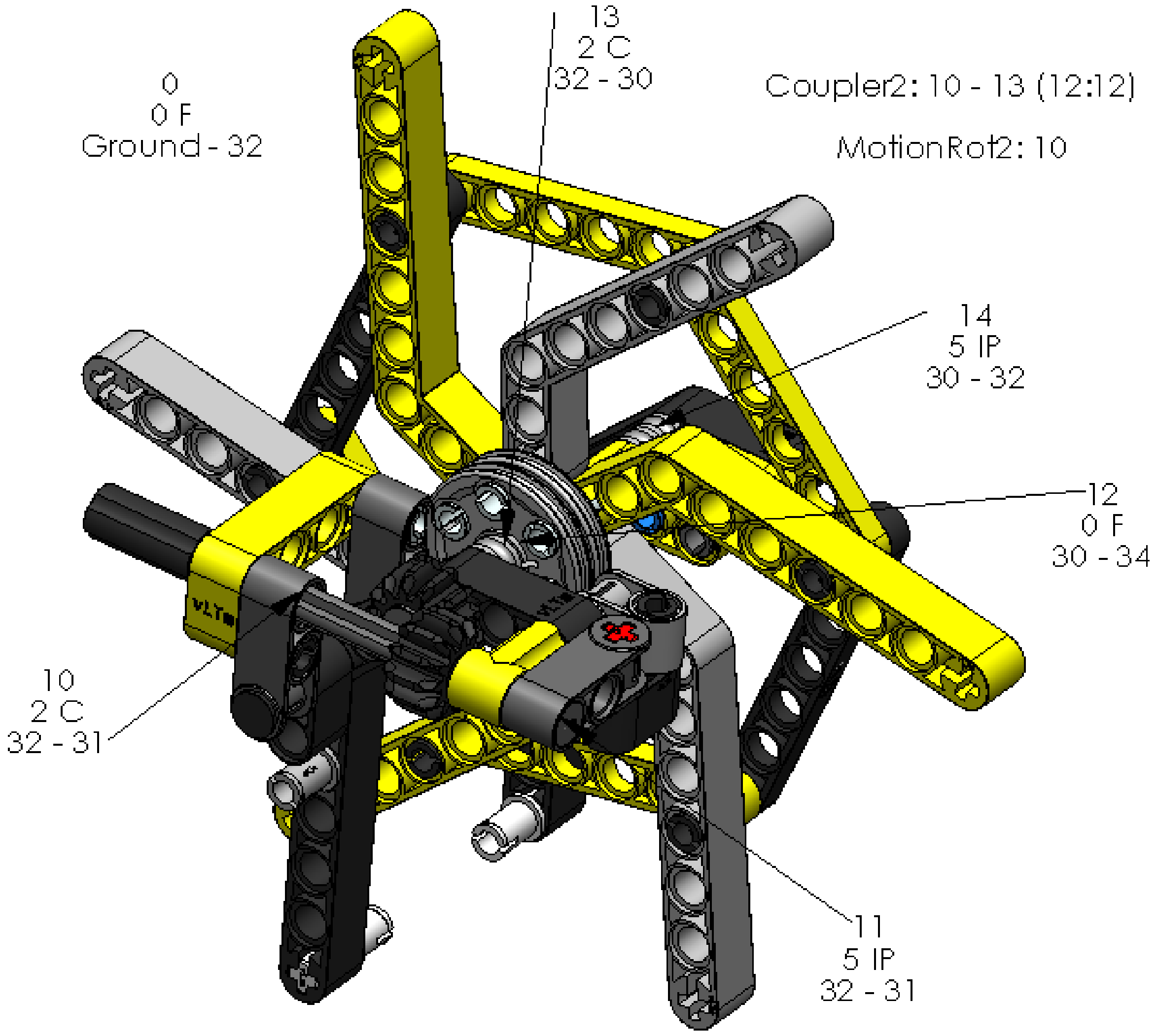


Figure 100. LTM_42028-2 >"W" > Self-aligned "Kinematic chain #1" in Recurdyn.



2nd KC:
Nb = 4 (32 is fixed; 30, 31, 34)
Nco = 5 (10, 11, 12, 13, 14; and 0). #0 and 12 are F
Sumat(f_i) = 0F*2 + 2C*2 + 5IP*2 + (-1)Cou*1 = 13
Sumat(6-f_i) = 6F*2 + 4C*2 + 1IP*2 + 1Cou*1 = 23

Map = 1
M3D = 6*(Nb-Nco-1) + Sumat(f_i) = 1
M3D = 6*Nb - Sumat(6-f_i) = 1

So, it is needed one M in joint #10

Figure 101. LTM_42028-2 > "W" > "Kinematic diagram of the 2nd Kinematic chain" in SolidWorks® 2020.

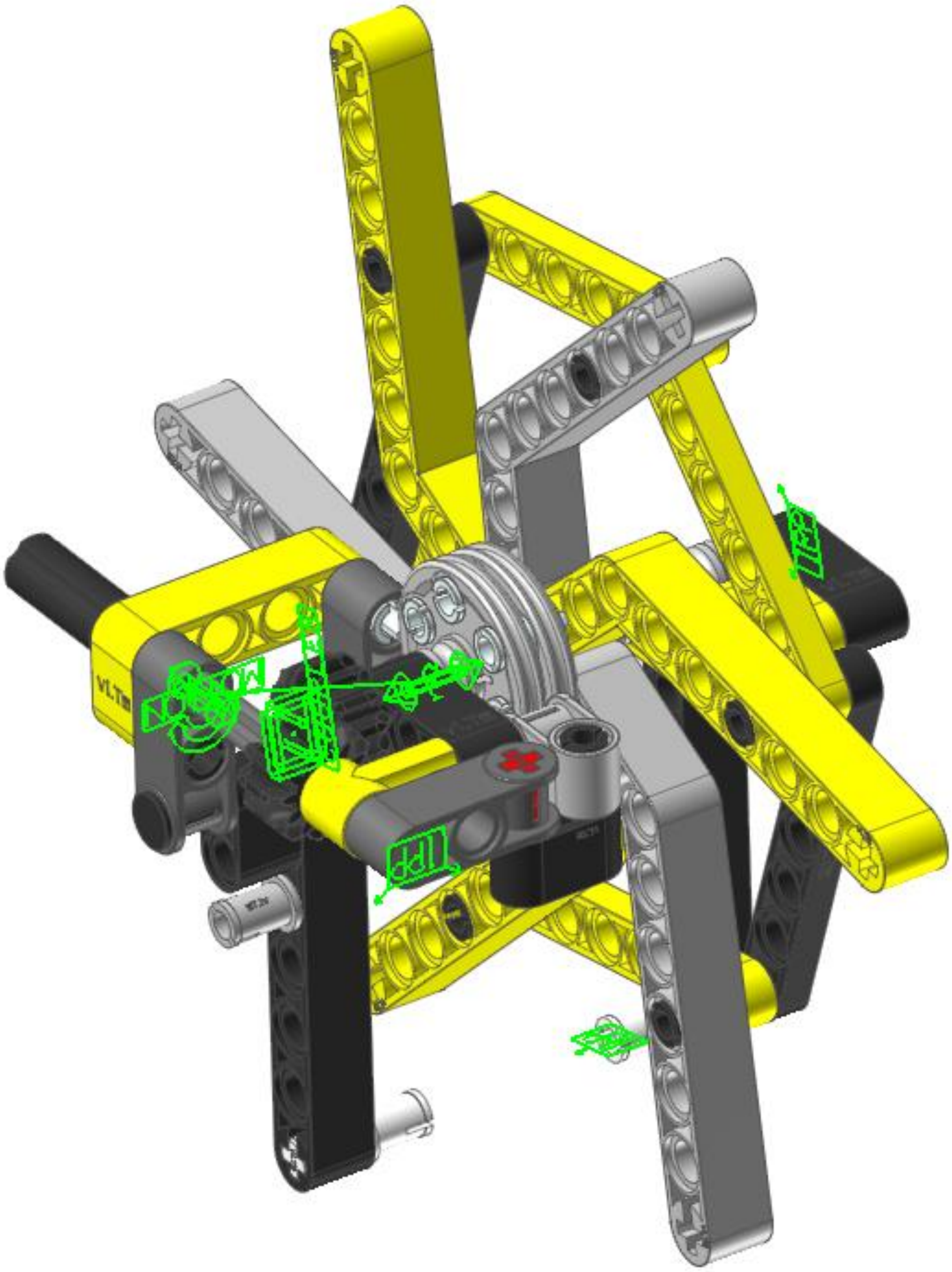


Figure 102. LTM_42028-2 > "W" > Self-aligned "Kinematic chain #2" in Recurdyn.

1.1.3. SIMULATION: VLTM_42028-2

vLTm_42028-2_2021 -wB-14 - w_contacts

vLTm_42028-2_2021 -wB-14 - w_contacts

vLTm_42028-2_2021 -wB-14 - w_contacts

1.2. MODEL #6: LTM_42043-2

1.2.1. ASSEMBLY: LTM_42043-2

Figure 103. LTM_42043-2 in real life. Source: this is the source. (www.jakexill.top/)

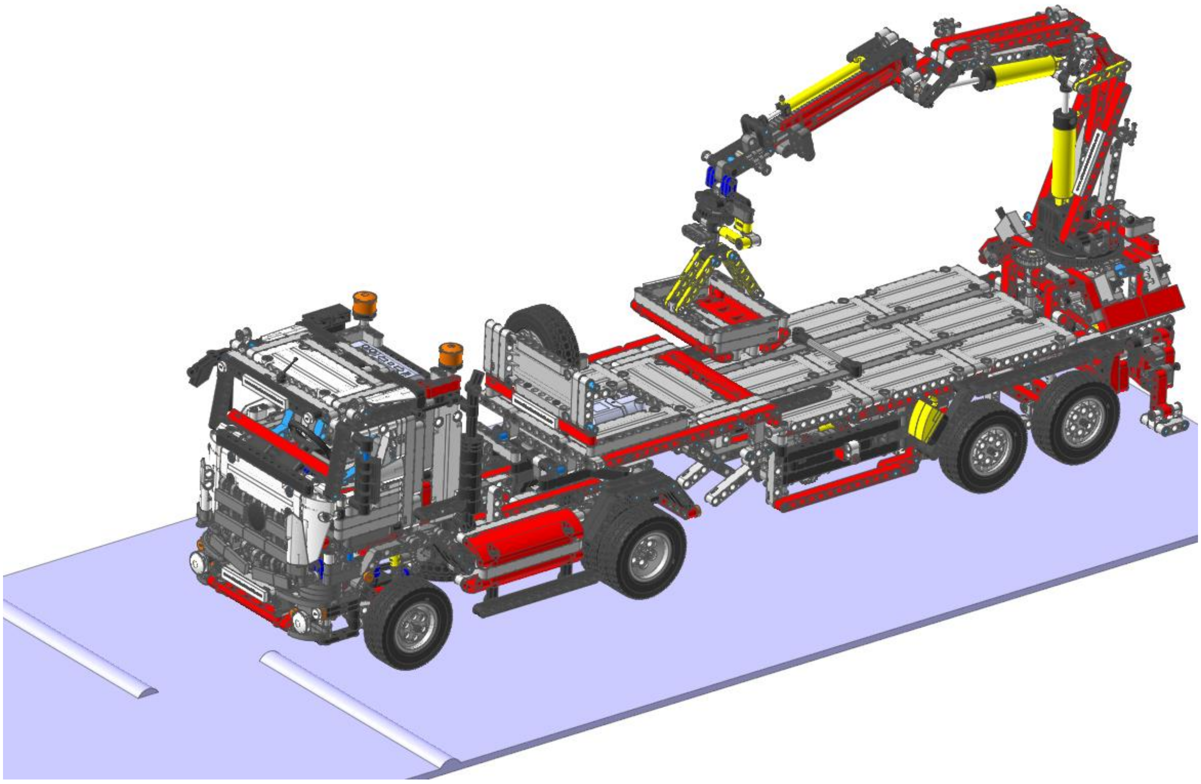


Figure 104. LTM_42043-2 in Recurdyn.

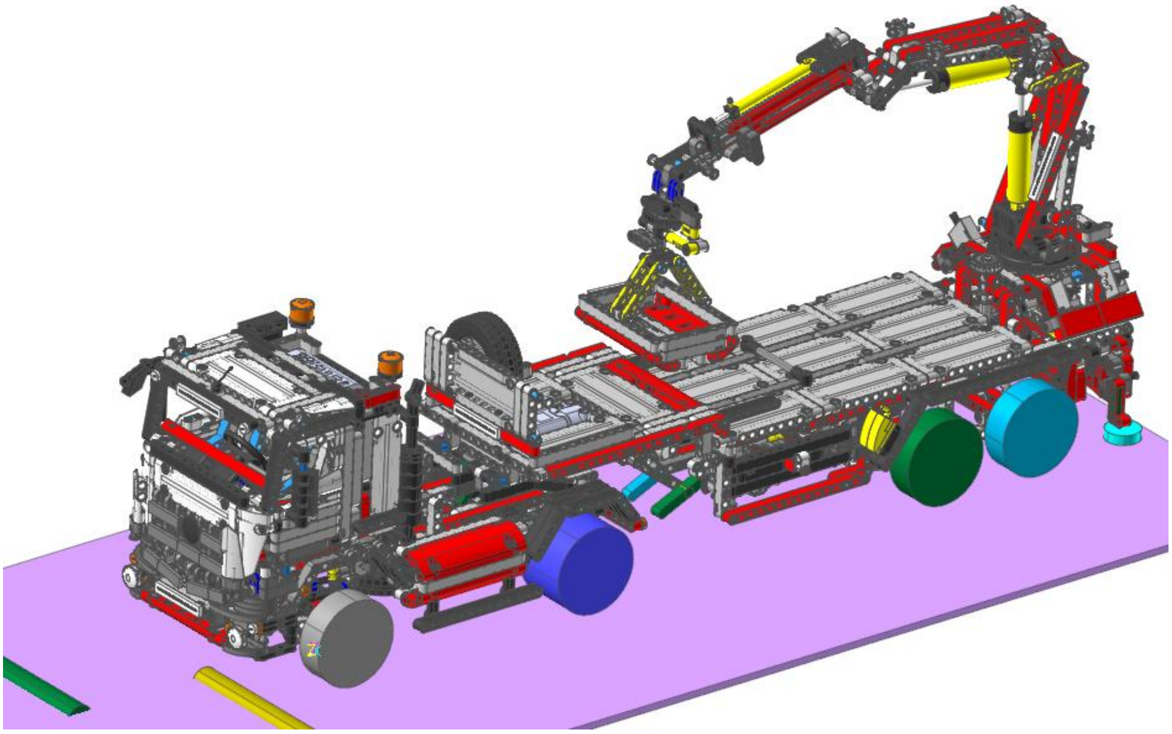


Figure 105. LTM_42043-2 in Recurdyn after using primitive bodies to reduce time simulation because of contact constraints.

*Figure 106. LTM_42043-2 > A real life machine (#1 > 1/1): MAN TGS 26.400 Skrzynia 6,25 m + HDS * 6x4. Source: this is the source (allegro.pl/)*

Figure 107. LTM_42043-2 > A real life machine (#2 > 1/2) > Trailer.

Figure 108. LTM_42043-2 > A real life machine (#2 > 2/2). > Truck.

Figure 109. LTM_42043-2 > A real life machine (#3 > 1/2): SCANIA R 480 / 6X4 / HDS PALFINGER PK 40002 + FLY JIB. Source: this is the source (matysek.pl/en)

Figure 110. LTM_42043-2 > A real life machine (#3 > 2/2): SCANIA R 480 / 6X4 / HDS PALFINGER PK 40002 + FLY JIB. Source: this is the source (matysek.pl/en)

Figure 111. LTM_42043-2 > A real-life machine (#4 > 1/3).

Figure 112. LTM_42043-2 > A real-life machine (#4 > 2/3).

Figure 113. LTM_42043-2 > A real-life machine (#4 > 3/3).

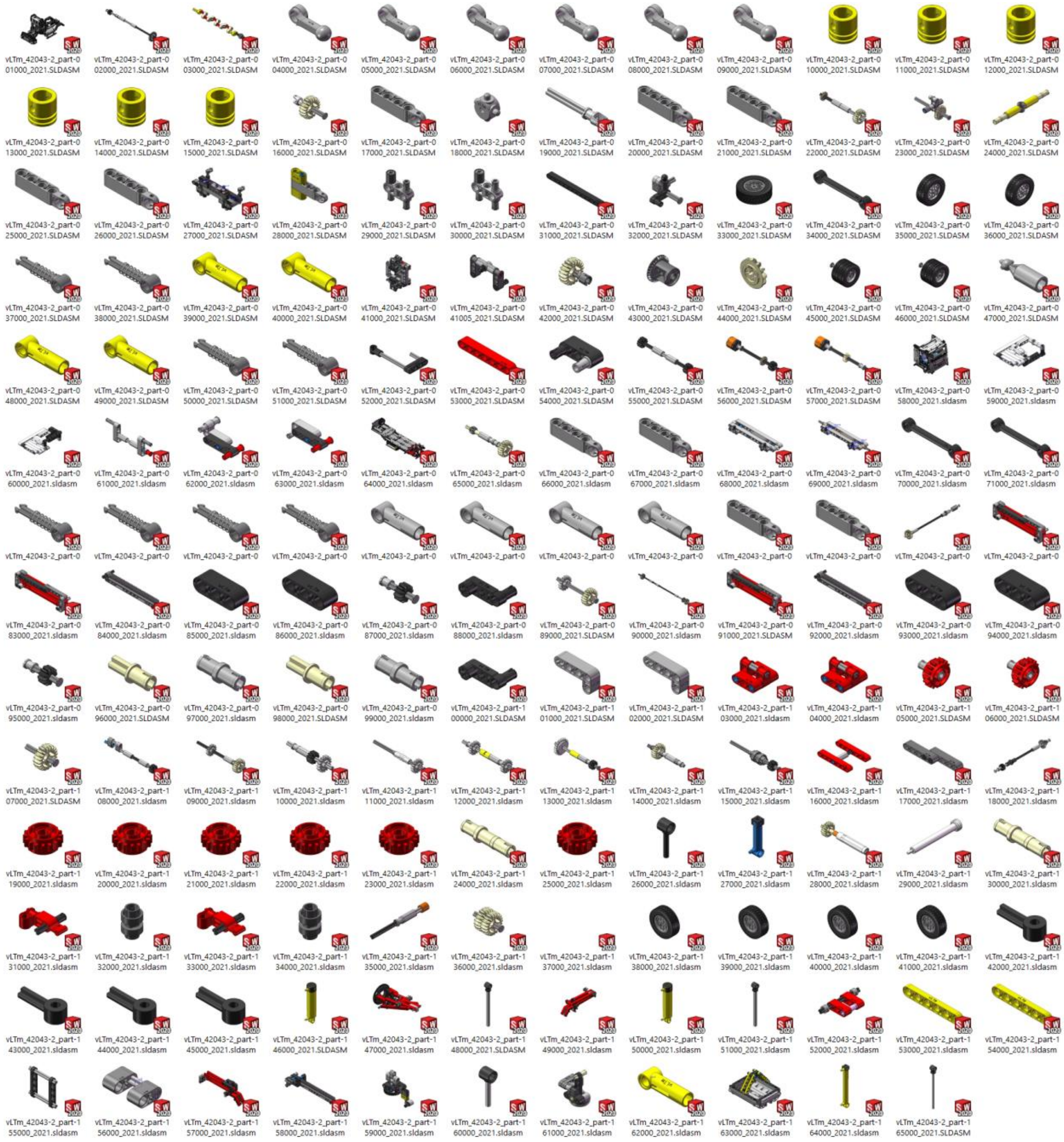


Figure 114. LTM_42043-2 > All the assemblies.

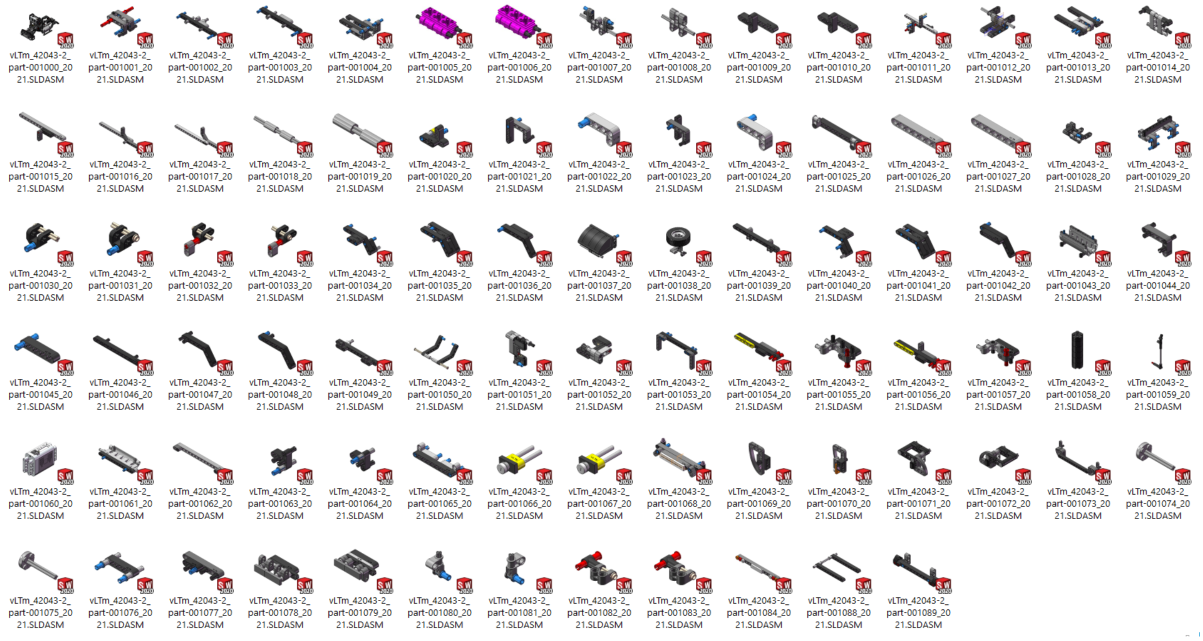


Figure 115. LTM_42043-2 > All parts of assembly nº 001.

Figure 116. LTM_42043-2 > All parts of assembly nº 058.

Figure 117. LTM_42043-2 > All parts of assembly nº 064.

Figure 118. LTM_42043-2 > All parts of assemblies nº 027, 041, 059, 060, 068, 069, 083, 091, 147, 149, 157, 158, 159, 161 and 163.

Figure 119. LTM_42043-2 > All the components.

1.2.2. SELF-ALIGNED MECHANISMS: VLTM_42043-2

1.2.2.1. MECHANISM “JOINT & ALL OPENED KINEMATIC CHAINS”, JA

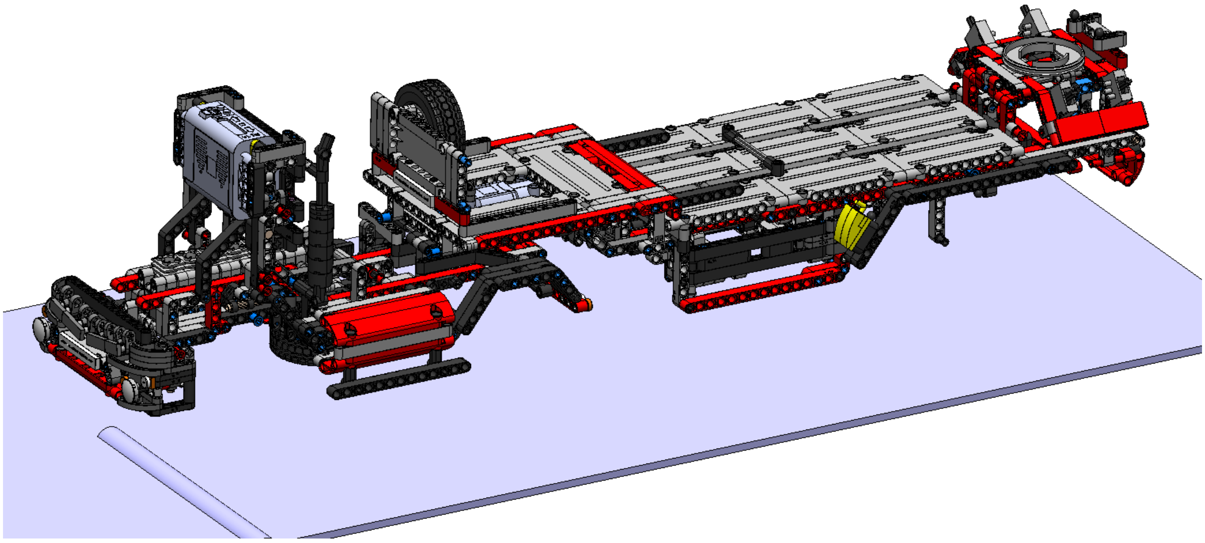


Figure 120. LTM_42043-2 > Mechanism “Joint & All opened kinematic chains“ (JA) in SolidWorks® 2020.

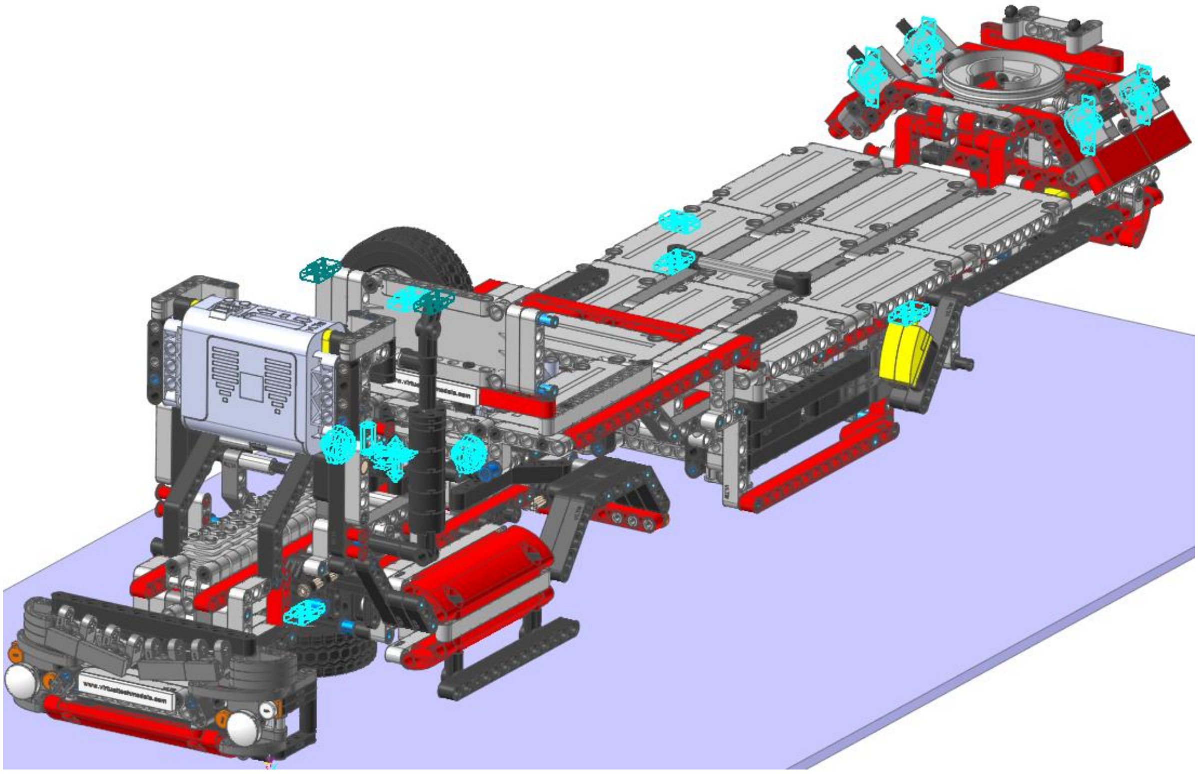


Figure 121. LTM_42043-2 > Self-aligned “JA” in Recurdyn.

vLTm_42043-2_part-001000_2021.SLDASM
vLTm_42043-2_part-033000_2021.SLDASM
vLTm_42043-2_part-052000_2021.SLDASM
vLTm_42043-2_part-053000_2021.SLDASM
vLTm_42043-2_part-054000_2021.SLDASM
vLTm_42043-2_part-064000_2021.sldasm
vLTm_42043-2_part-101000_2021.SLDASM
vLTm_42043-2_part-130000_2021.sldasm
vLTm_42043-2_part-137000_2021.sldasm
vLTm_42043-2_part-142000_2021.sldasm
vLTm_42043-2_part-143000_2021.sldasm
vLTm_42043-2_part-144000_2021.sldasm
vLTm_42043-2_part-145000_2021.sldasm

Figure 122. LTM_42043-2 > “JA” > All the assemblies.

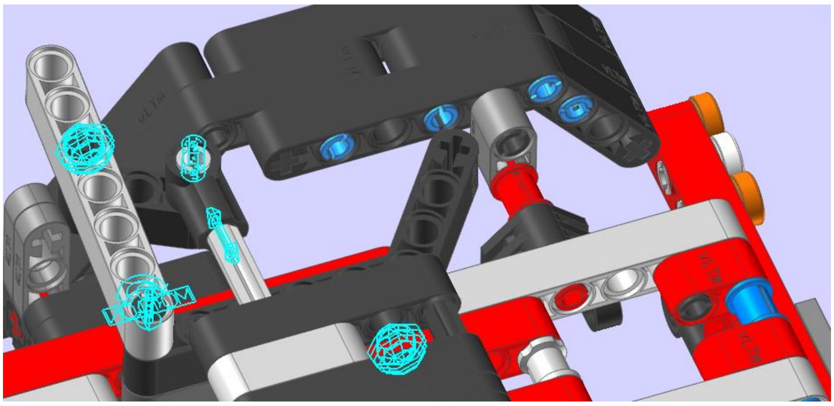
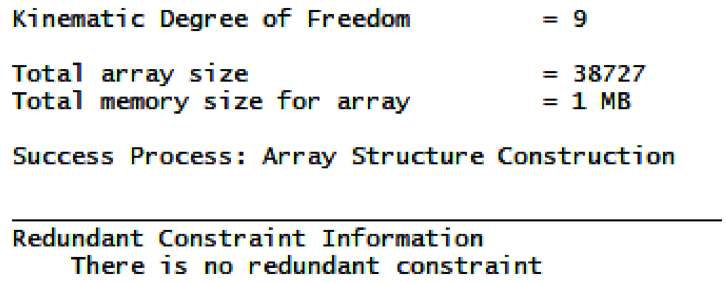


Figure 123. LTM_42043-2 > Self-aligned “JA” > A detail of the joint between truck and trailer in Recurdyn.



```
Kinematic Degree of Freedom        = 9

Total array size                   = 38727
Total memory size for array        = 1 MB

Success Process: Array Structure Construction

________________________________________________
Redundant Constraint Information
    There is no redundant constraint
```

Figure 124. LTM_42043-2 > “JA” has 9 degrees of freedom: truck 001 has 6 as free body, and trailer 064 has 3 because of spherical joint to 001. Also, there is no redundant constraints according to Recurdyn > Control panel “Message”.

1.2.2.2. MECHANISM “CABIN”, C

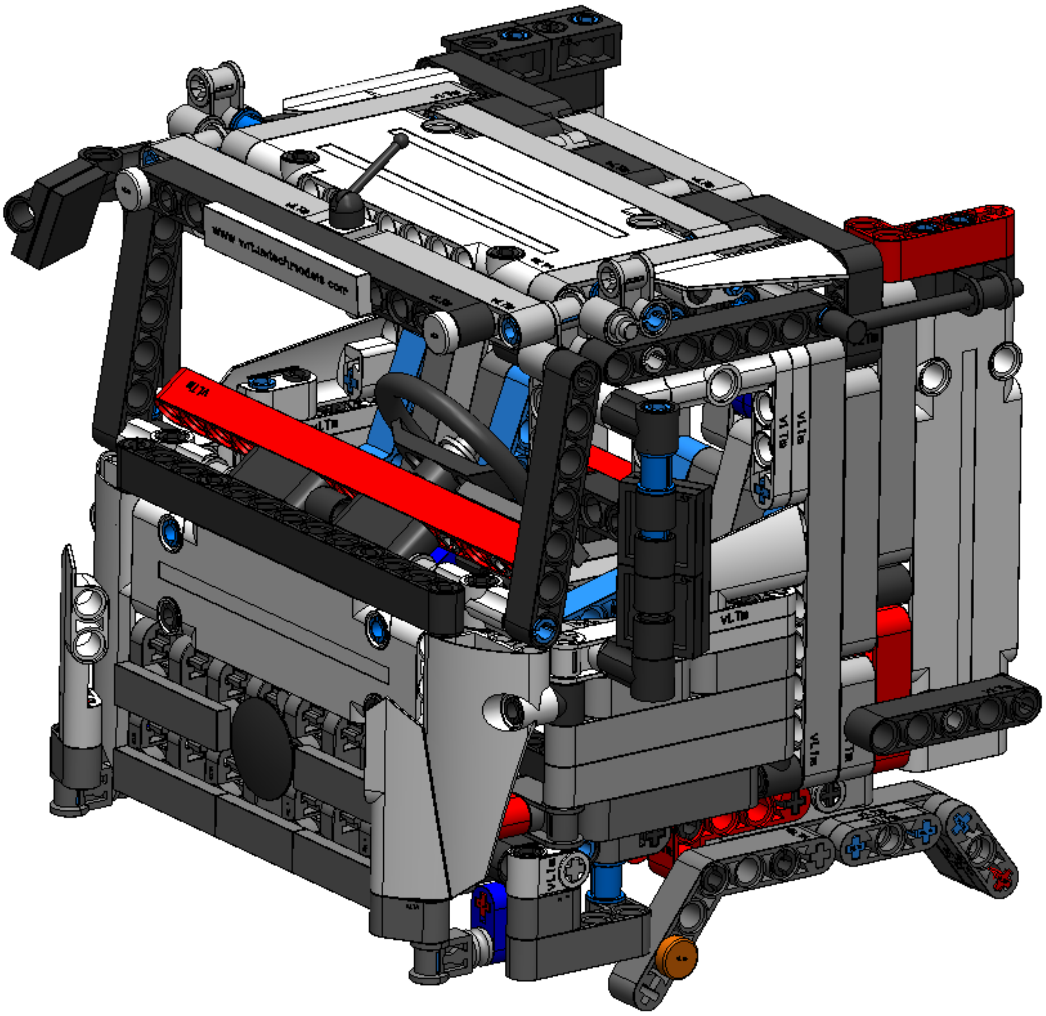
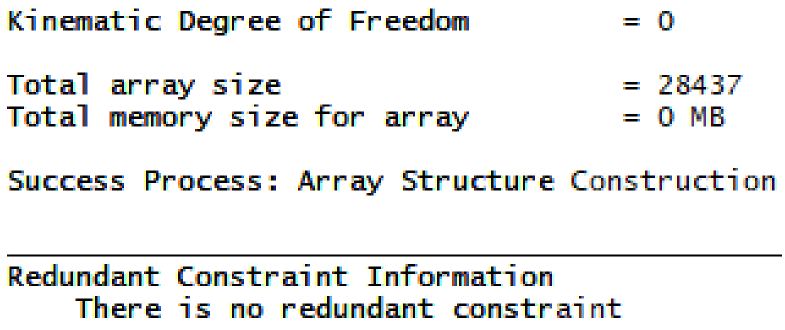


Figure 125. LTM_42043-2 > Mechanism “Cabin” (C) in SolidWorks® 2020. FTTG: 001.



```
Kinematic Degree of Freedom         = 0

Total array size                    = 28437
Total memory size for array         = 0 MB

Success Process: Array Structure Construction

Redundant Constraint Information
    There is no redundant constraint
```

Figure 126. LTM_42043-2 > “C” has 4 rotational degrees of freedom to be controlled: 2 doors, the steering wheel, and the up-and-down movement of the cabin. In the panel appears 0 because a motion in the properties of each joint has been created for the simulation.

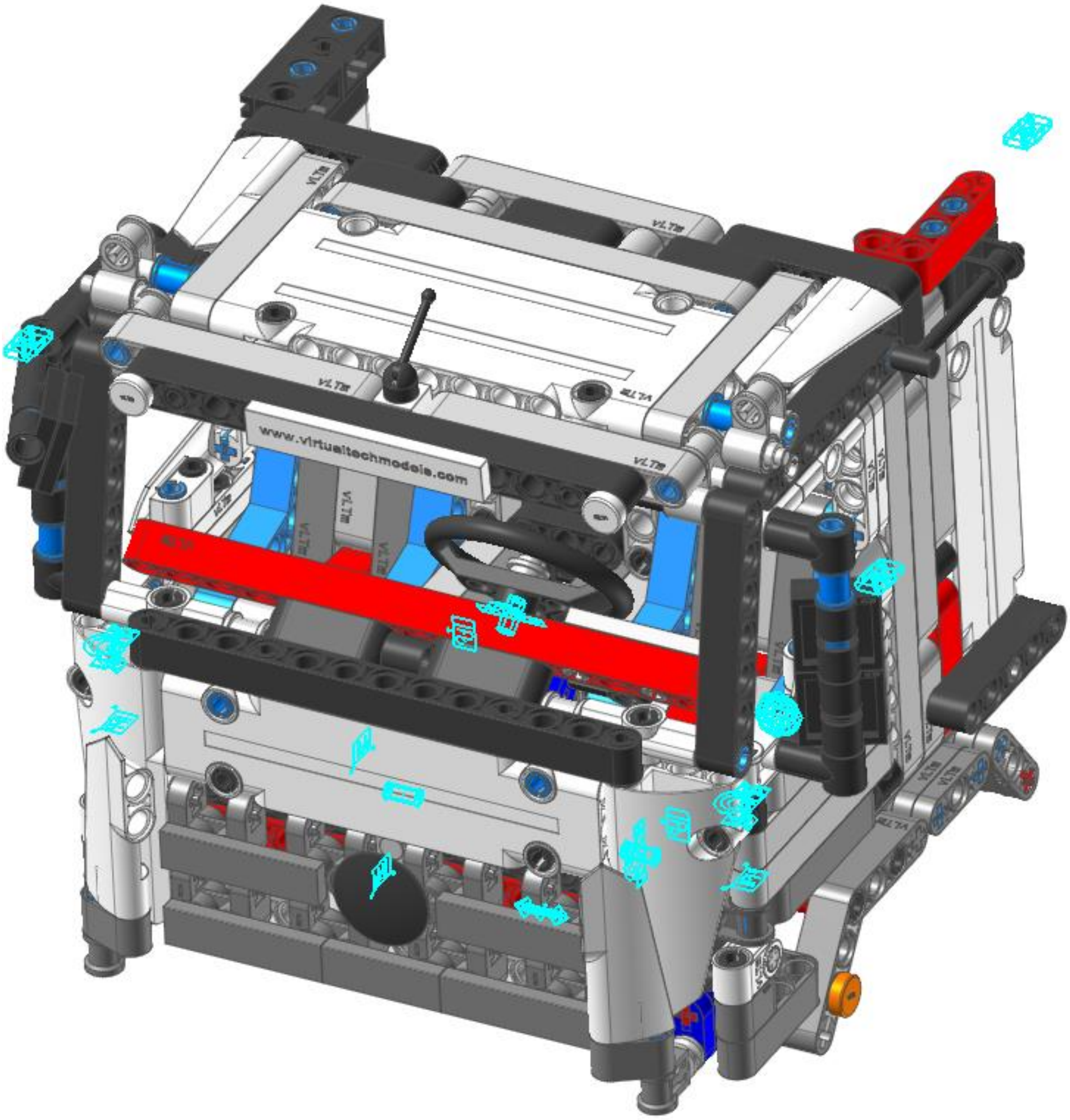


Figure 127. LTM_42043-2 > Self-aligned "C" in Recurdyn.

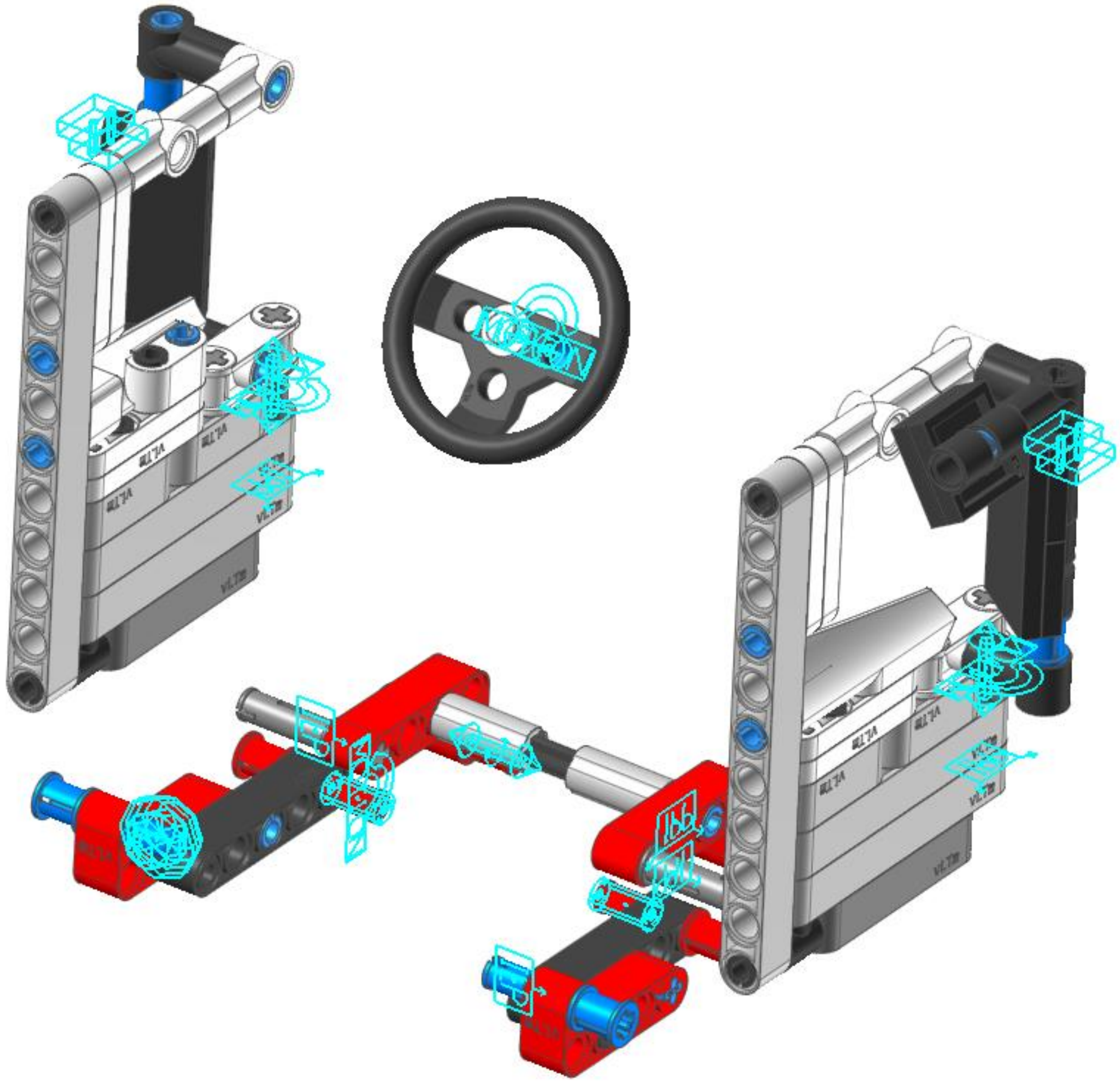


Figure 128. LTM_42043-2 > Self-aligned "C" with piece nº 058 hidden in Recurdyn.

Figure 129. LTM_42043-2 > “C” > All the assemblies.

1.2.2.3. MECHANISM “DIRECTION”, D

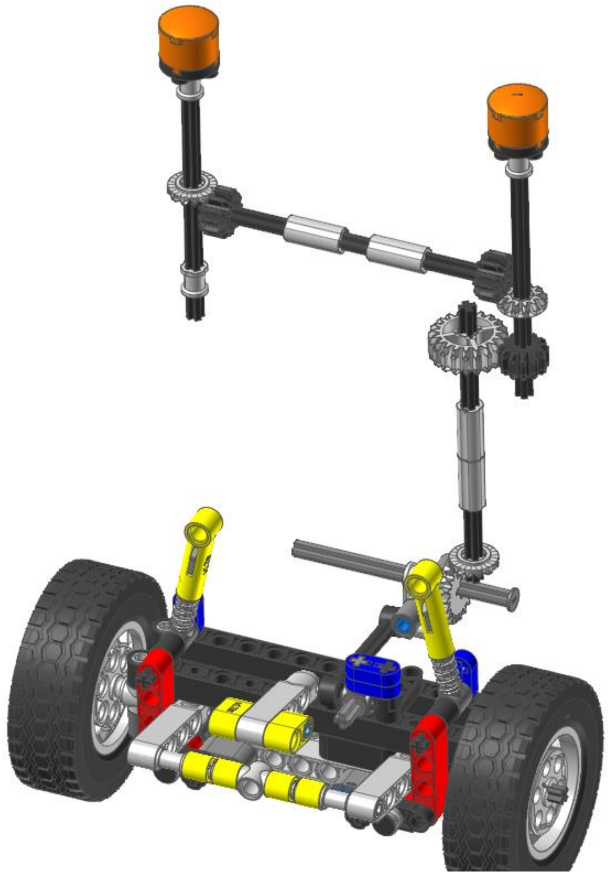


Figure 130. LTM_42043-2 > Mechanism “Direction” (D) in SolidWorks® 2020. FTTG: 001

vLTm_42043-2_part-022000_2021.SLDASM
vLTm_42043-2_part-023000_2021.SLDASM
vLTm_42043-2_part-024000_2021.SLDASM
vLTm_42043-2_part-025000_2021.SLDASM
vLTm_42043-2_part-026000_2021.SLDASM
vLTm_42043-2_part-027000_2021.SLDASM
vLTm_42043-2_part-028000_2021.SLDASM
vLTm_42043-2_part-029000_2021.SLDASM
vLTm_42043-2_part-030000_2021.SLDASM
vLTm_42043-2_part-031000_2021.SLDASM
vLTm_42043-2_part-032000_2021.SLDASM
vLTm_42043-2_part-034000_2021.SLDASM
vLTm_42043-2_part-035000_2021.SLDASM
vLTm_42043-2_part-036000_2021.SLDASM
vLTm_42043-2_part-037000_2021.SLDASM
vLTm_42043-2_part-038000_2021.SLDASM
vLTm_42043-2_part-039000_2021.SLDASM
vLTm_42043-2_part-040000_2021.SLDASM
vLTm_42043-2_part-055000_2021.SLDASM
vLTm_42043-2_part-056000_2021.SLDASM
vLTm_42043-2_part-057000_2021.SLDASM

Figure 131. LTM_42043-2 > “D” > All the assemblies.

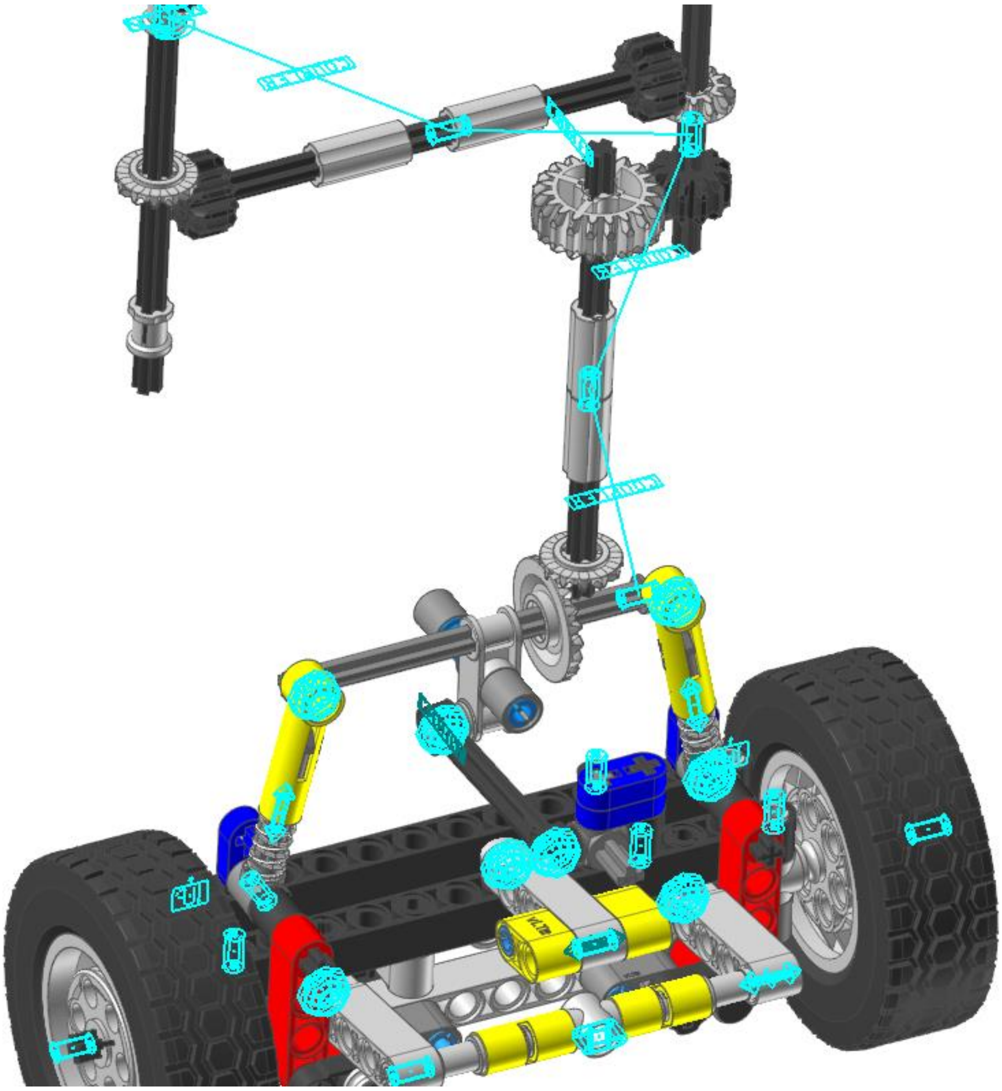
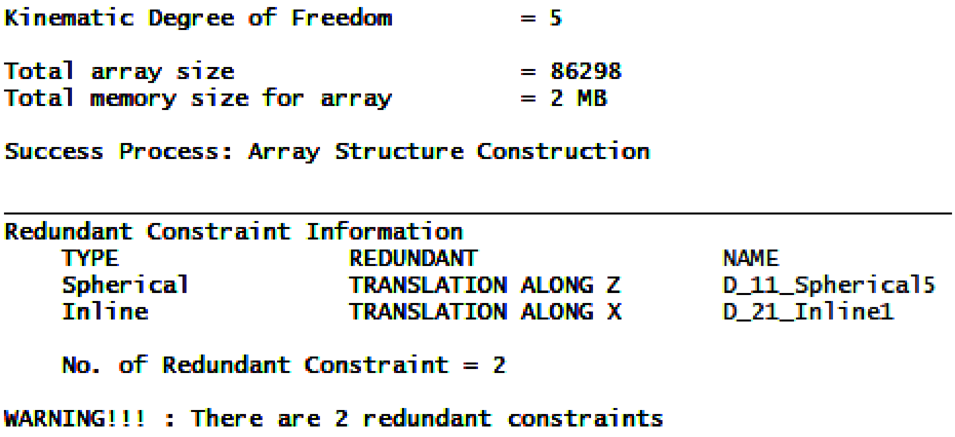


Figure 132. LTM_42043-2 > ***NO*** *self-aligned "D" in Recurdyn. To recreate shock absorbers motion, springs (forces) are defined.*



```
Kinematic Degree of Freedom          = 5

Total array size                     = 86298
Total memory size for array          = 2 MB

Success Process: Array Structure Construction

Redundant Constraint Information
    TYPE                REDUNDANT                 NAME
    Spherical           TRANSLATION ALONG Z       D_11_Spherical5
    Inline              TRANSLATION ALONG X       D_21_Inline1

    No. of Redundant Constraint = 2

WARNING!!! : There are 2 redundant constraints
```

Figure 133. LTM_42043-2 > "D" should have 5 degrees of freedom: 2 non-driving wheels, 2 independent shock absorbers and 1 movement to go to the left or right.

In this mechanism there are some problems because of design and arrangement of pieces. Next, it is explained.

1. These are the DOF of the simulated mechanism:

- 2 non-driving wheels, so +2 DOF. In the simulated mechanism is okey
- 1 independent movement for both shock absorbers, so +1 DOF. It should be +two DOF because each shock absorber can have a different motion with respect to the other one. In the simulated mechanism is wrong.

  In the next figure, the red joints are activated; and the dark blue joint is not. In the axis where there are two joints, there are 1 R and 1 C joint. It lets to there be +one DOF. If both were R, it would be over-constrained because the second R would not be constraining a different DOF with respect to the first R. So, to get the needed +two DOF, the top red joint should be a C joint. However, if this were the case, this piece could move along its axis because there is no R joint to constraint said movement. So, to make both shock absorbers work independently is not possible because of the design created by LEGO®.

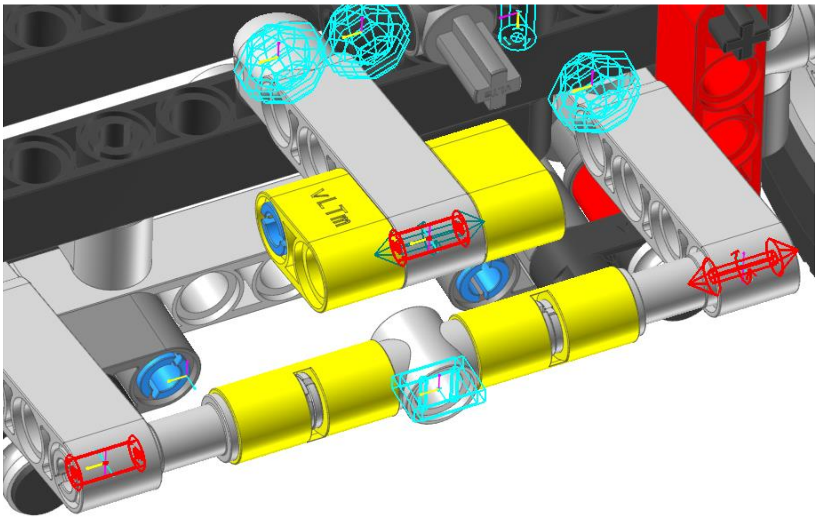


Figure 134. LTM_42043-2 > "D" > Detail 1/5.

- 1 movement to go to the left or right. In the simulated mechanism it is okey, but since it is already controlled, in the control panel this DOF Is not supposed to appear, so +0 DOF
- The black piece placed in the middle in the figure can rotate along its axis because it is constrained by two spherical joints. It also happens with the shock absorber placed at the right side of the figure. So, +2 DOF. In the simulated mechanism it is wrong. It should be + zero DOF. To mathematically delete the last two DOF, a constraint called "CMotion" can be defined in Recurdyn. However, visually can be reduced by defining friction in one or both spherical joints of each piece.

2. The two redundant constraints cannot be eliminated either. After the following figures, it is the explication.

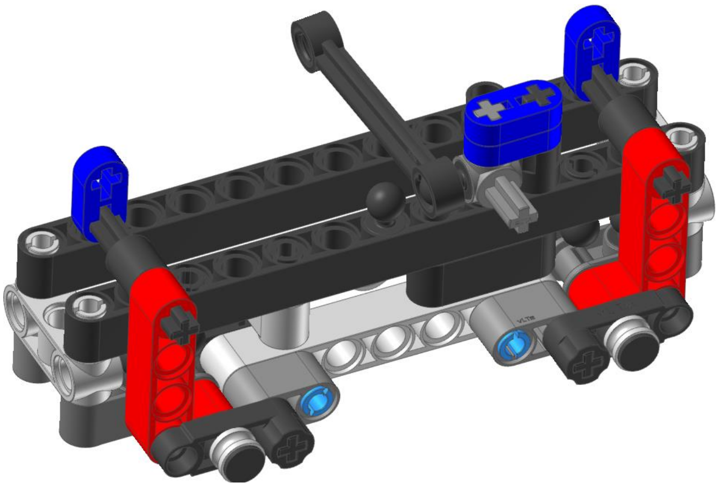


Figure 135. LTM_42043-2 > "D" > Detail 2/5.

vLTm_42043-2_part-027000_2021.SLDASM

vLTm_42043-2_part-029000_2021.SLDASM

vLTm_42043-2_part-030000_2021.SLDASM

vLTm_42043-2_part-031000_2021.SLDASM

vLTm_42043-2_part-032000_2021.SLDASM

vLTm_42043-2_part-034000_2021.SLDASM

Figure 136. LTM_42043-2 > "D" > Detail 3/5.

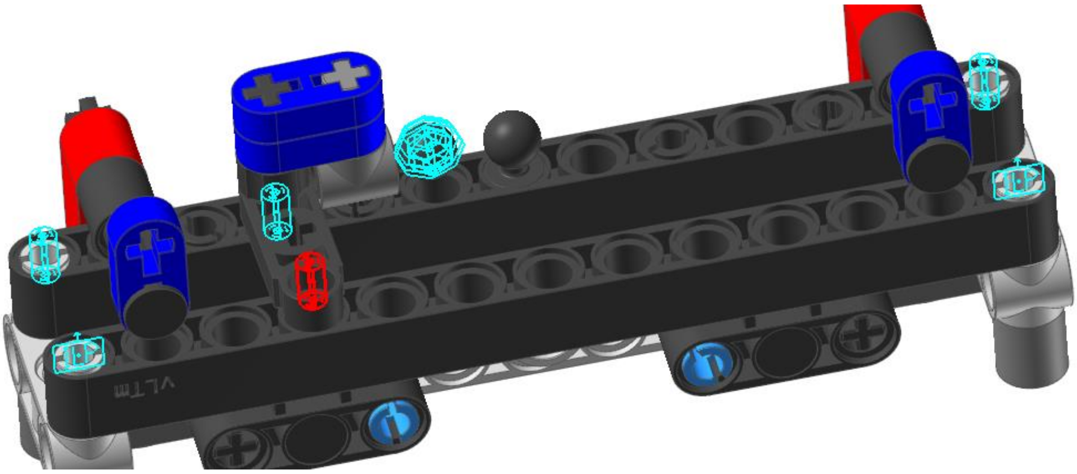


Figure 137. LTM_42043-2 > "D" > Detail 4/5.

Piece nº 027 is going to be considered as fixed to the ground. There are 5 moving parts. When 034 is moved because of some movement from gears (from 023), 032 is also moved. When 032 is moved, 031 is also moved because they are jointed. When 031 is moved, 029 and 030 are moved with respect to 027 (R joints) and 031 (IL joints). All movements must be rotational movement, except the one from the Sph joint. Thus, after defining all the joints, pieces should only be able to rotate. They cannot have a translational movement. To make it impossible, a R joint must be defined. This way, by defining all blue R joint it is achieved. But 031 is not already constrained, so between 031 and 032 the red R joint need to be defined. Now, just need to make 029 and 030 follow the movement of 031. If 2 IL are defined, 2 redundant constraints appear; so, if 2 C, 2*(f_i(IL) – f_i(C)) = 2*(4_{IL} – 2_C) = 2*2 = 4 redundant constraints appear; if 2 R, 6; if 2 IP or 2 Sph, 029 and 030 are not concentric with the corresponding hole. So the joints to be defined are two IL joints, because they create less redundant constrains than two R joint and because other options are not possible. *Note: it is calculated with f_i(…), not with 6 – f_i(…), because the more practice equation is M_{3D}(f_i).*

It also can be seen from a mathematical point of view. Since this whole subsystem only has 1 DOF, 034 is not included in the subsystem because the motion can be defined in the blue R joint of 032, and it is wanted to be self-aligned, M_{ap} = M_{3D}, 1 = 6(5-A-1)+(1_R*4+x), where A=6 is the nº of joints (the 3 blue + 1 red + 2 joints "@" are left to be defined). Solving, x= +9 DOF. To say that A=3+1+2=6 is the same as saying that A = 1_R*3+1_R*1+1_R*2 because the DOF of a R joint is 1. This way, A = 3+1+2 = 1*3+1*1+1*2 = 1_R*3+1_R*1+1_R*2 = 6. So, to say that 1_R*4+x is the nº of total DOF the same as saying that there are (1_R*4+x)=13 R joints, because 1 R joint lets 1 DOF. In consequence, "@"="R". This way, by changing the two "@", which can be considered as R joints, a positive increment of +9 DOF has to be achieved. If changing the two "@" joints to … C, 2*(2_C – 1_R) = +2; Sph, 2*(3_{Sph} – 1_R) = +4; IL, 2*(4_{IL} – 1_R) = +6; IP, 2*(5_{IP} – 1_R) = 8. As it shows, the maximum DOF to be augmented is 8. So, it is impossible to delete all constraint of the mechanism "D"; and the best option is to choose two IL joints: "@ ' " = "IL", because IP does not make a concentricity

After trying to self-aligning all this, NRC = 2. It also must be highlighted that:

- to recreate a simulation where the vehicle seems to move properly shown constraints are enough, but they do not perfectly recreate the motion of a real industrial vehicle.
- to self-align this mechanism "D", to change pieces and their design must be changed
- with the given pieces to be assembled, it is not always possible to get a self-aligned mechanism. This must be considered when designing a mechanical product.

Next, in the middle of the figure, can be seen how the piece 034 is jointed to piece 027 in a real machine:

Figure 138. LTM_42043-2 > "D" > Detail 5/5.

1.2.2.4. MECHANISM “POWER”, P

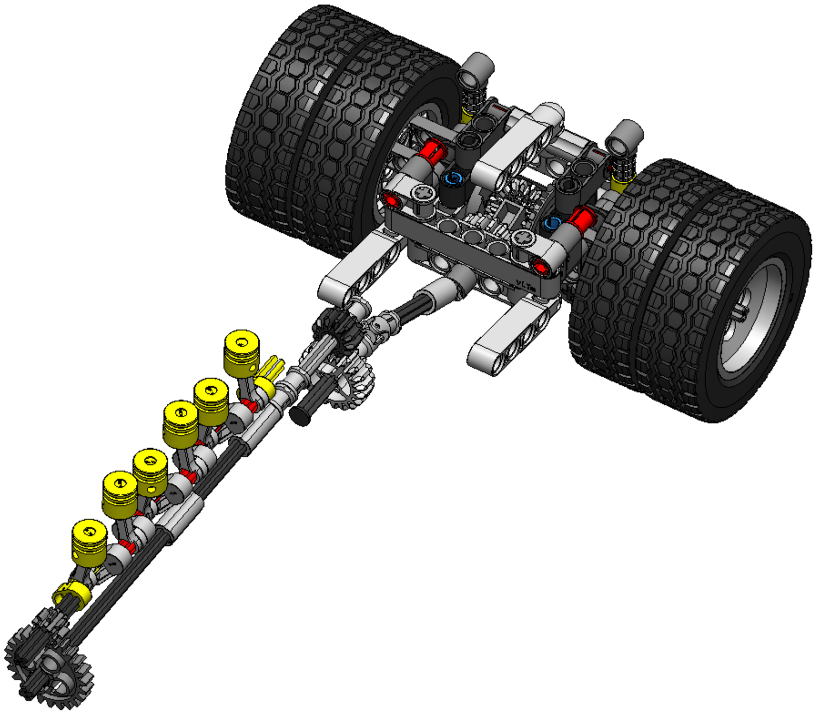


Figure 139. LTM_42043-2 > Mechanism “Power” (P) in SolidWorks® 2020. FTTG: 001.

Figure 140. LTM_42043-2 > “P” > All the assemblies.

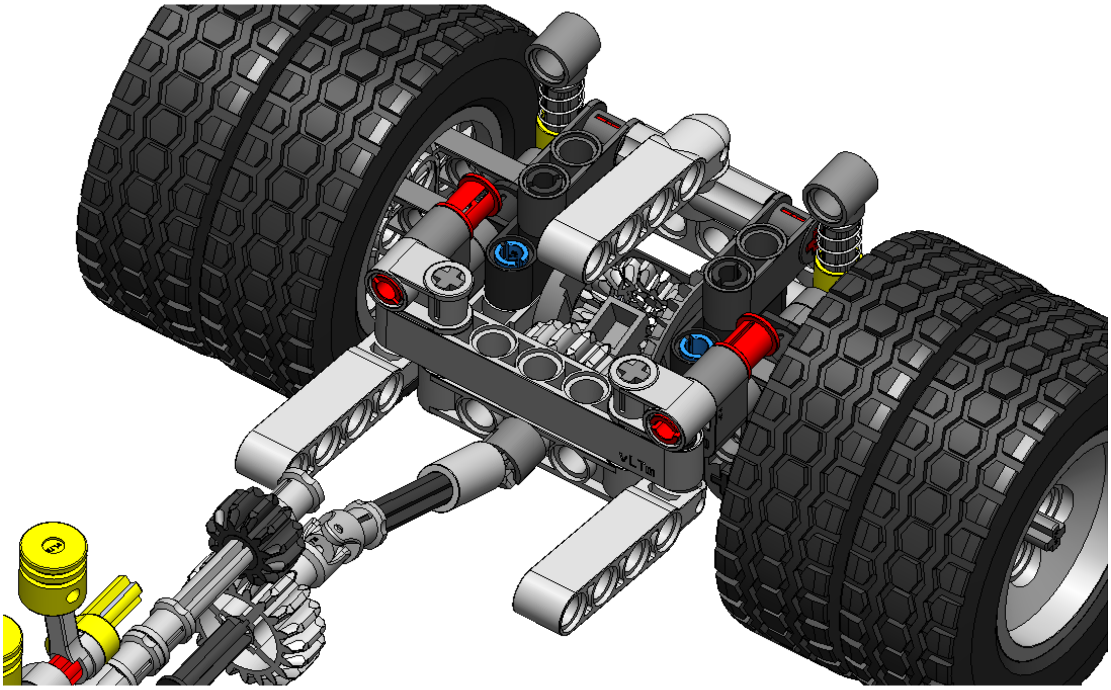


Figure 141. LTM_42043-2 >"P" > A detail of the differential and damping system in SolidWorks® 2020.

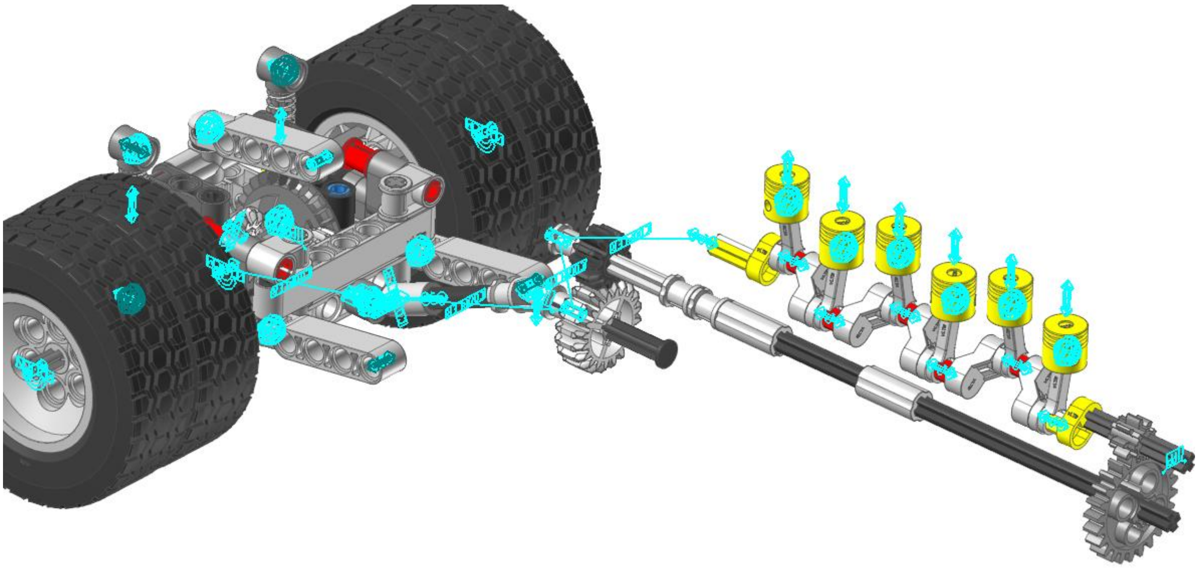
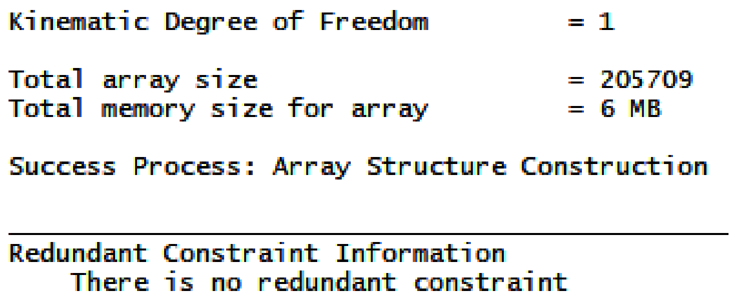


Figure 142. LTM_42043-2 > Self-aligned "P" in Recurdyn.



```
Kinematic Degree of Freedom          = 1

Total array size                     = 205709
Total memory size for array          = 6 MB

Success Process: Array Structure Construction

_______________________________________________
Redundant Constraint Information
    There is no redundant constraint
```

Figure 143. LTM_42043-2 > "P" should have 4 degrees of freedom: 2 controlled driving wheels (or 1 controlled crankshaft) and 2 independent shock absorbers (the simulated model only has 1 because the same problem than in "D").

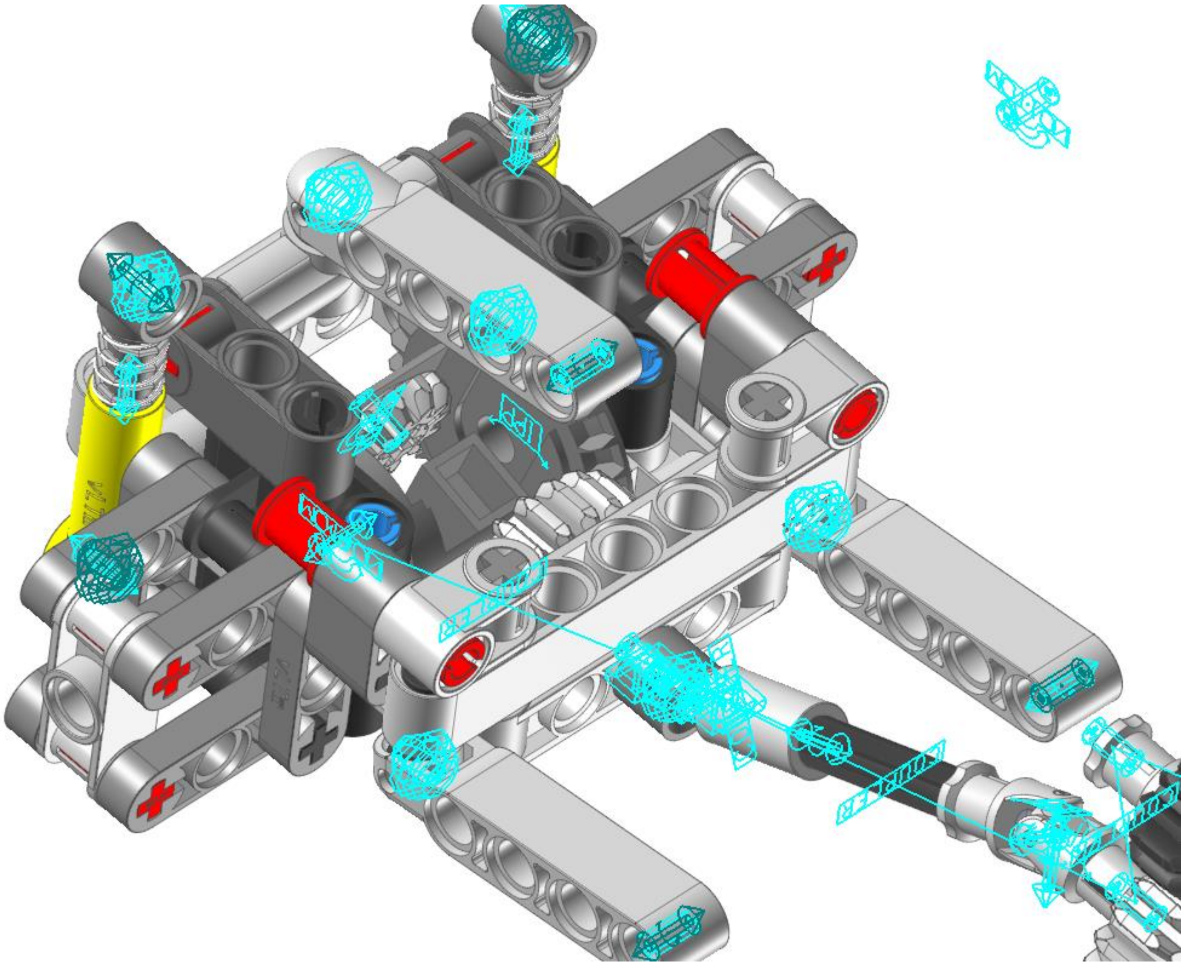


Figure 144. LTM_42043-2 >"P" > A detail of the differential and damping system in Recurdyn.

Figure 145. LTM_42043-2 >"P" > An example of the differential and the universal joint (1/2).

Figure 146. LTM_42043-2 >"P" > An example of the differential and the universal joint (2/2).

1.2.2.5. MECHANISM “TRAILER’S WHEELS”, W

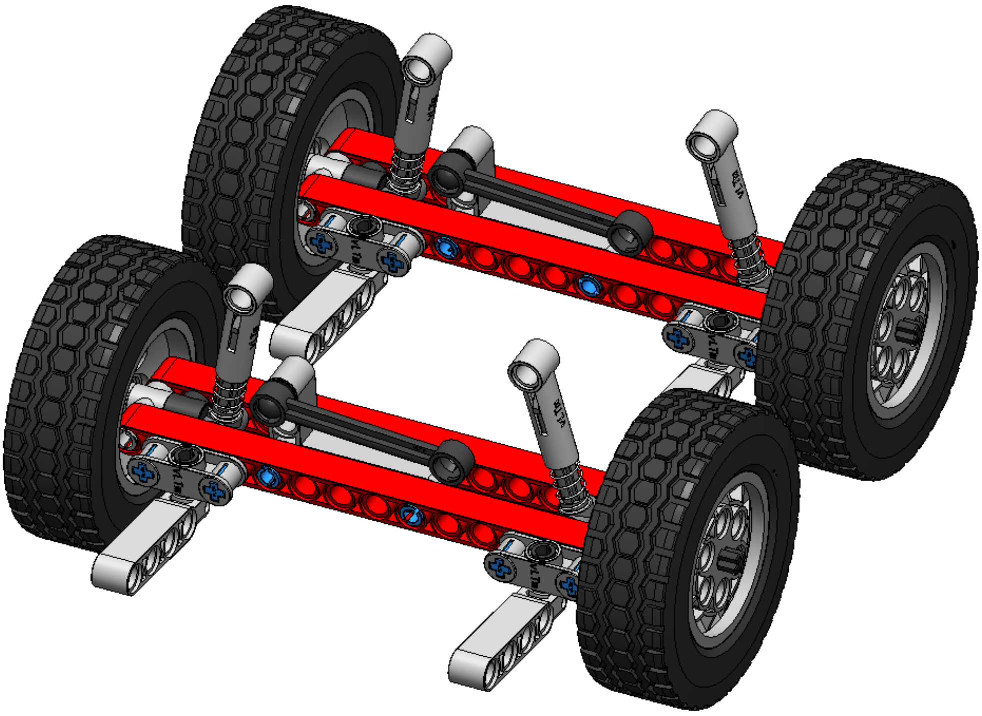


Figure 147. LTM_42043-2 > Mechanism “Trailer’s wheels” (W) in SolidWorks® 2020. FTTG: 064.

Figure 148. LTM_42043-2 > “W” > All the assemblies.

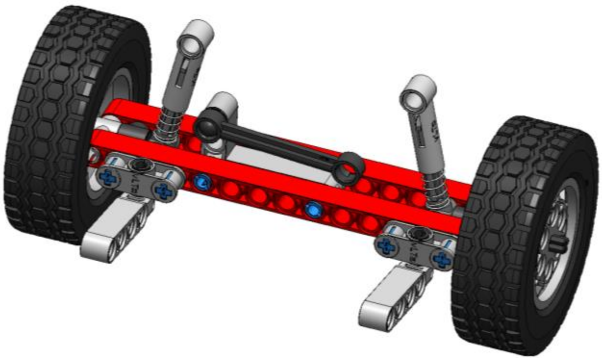


Figure 149. LTM_42043-2 > “W > A detail of the two equal kinematic chains.

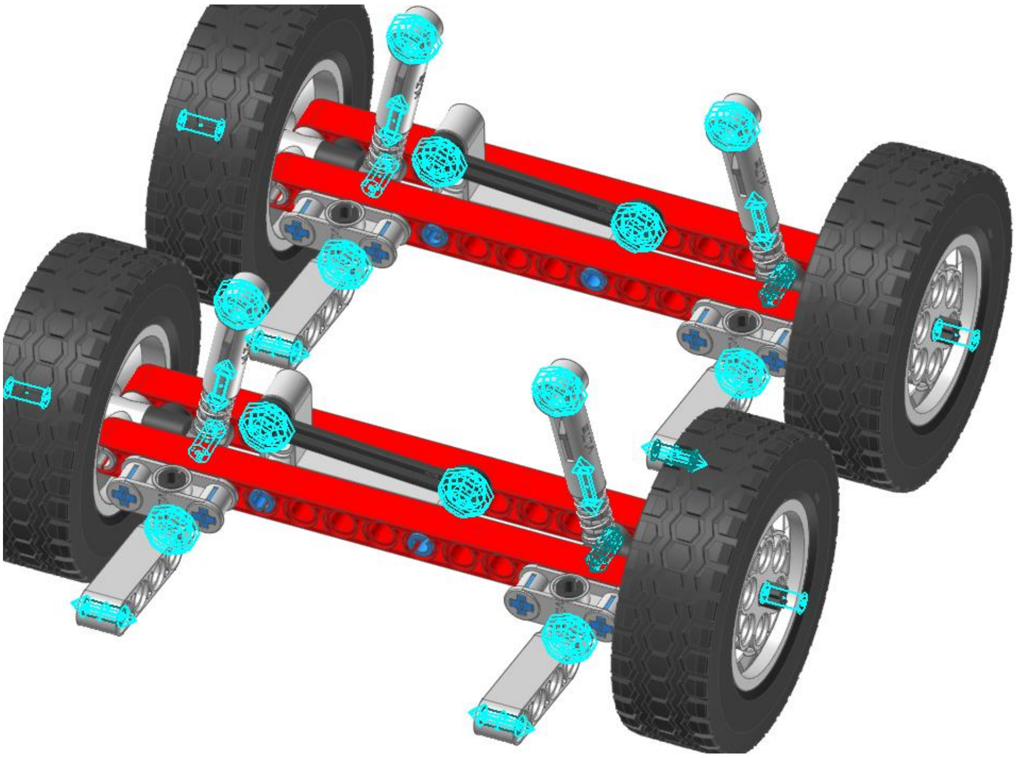
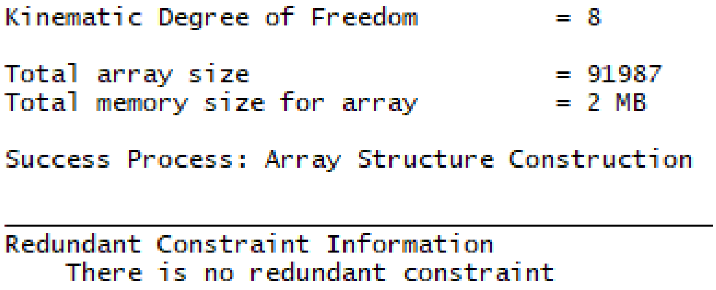


Figure 150. LTM_42043-2 > Self-aligned”W” in Recurdyn.



```
Kinematic Degree of Freedom          = 8

Total array size                     = 91987
Total memory size for array          = 2 MB

Success Process: Array Structure Construction

________________________________________________
Redundant Constraint Information
    There is no redundant constraint
```

Figure 151. LTM_42043-2 > “W” should have 8 no-controlled degrees of freedom: 4 non-driving wheels and 4 independent shock absorbers (the simulated model only has 2 because the same problem than in “D”; also, it has 2 extra because of two pieces constrained by 2 spherical joint each one).

1.2.2.6. MECHANISM “GEARS”, G

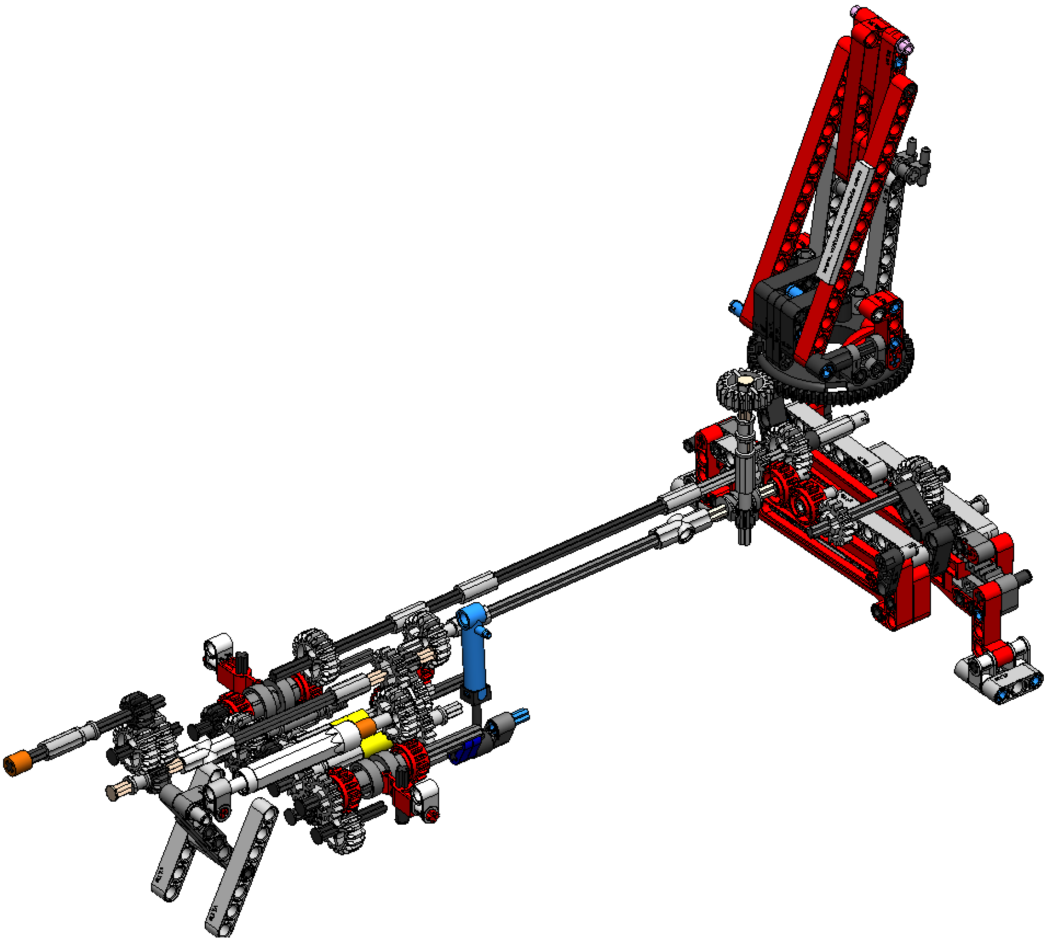


Figure 152. LTM_42043-2 > Mechanism “Gears” (G) in SolidWorks® 2020. FTTG: 064.

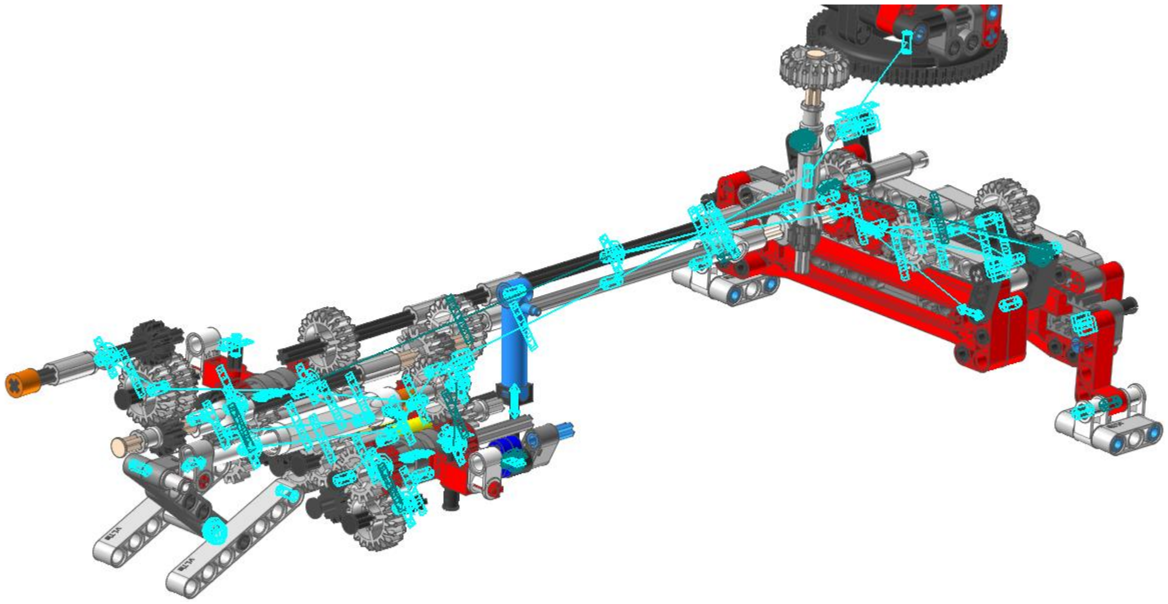
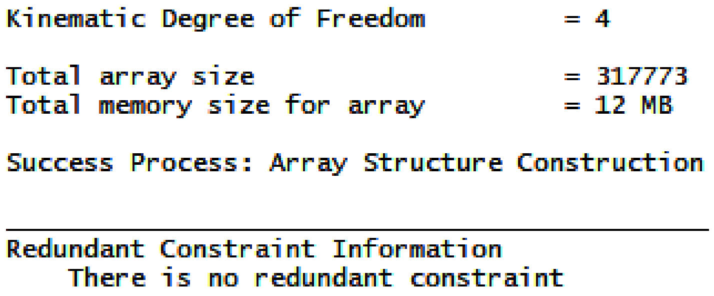


Figure 153. LTM_42043-2 > Self-aligned “G” in Recurdyn.



```
Kinematic Degree of Freedom          = 4

Total array size                     = 317773
Total memory size for array          = 12 MB

Success Process: Array Structure Construction

Redundant Constraint Information
    There is no redundant constraint
```

Figure 154. LTM_42043-2 > Self-aligned Mechanism "Gears" (G) has 8 DOF: 4 are to be controlled (the pump, the parking, the rotation of the crane, the extension or retraction of the supports) and the rest not (2 free rotation of supports, and the 2 big red back pieces has a no-desmodromic movement)

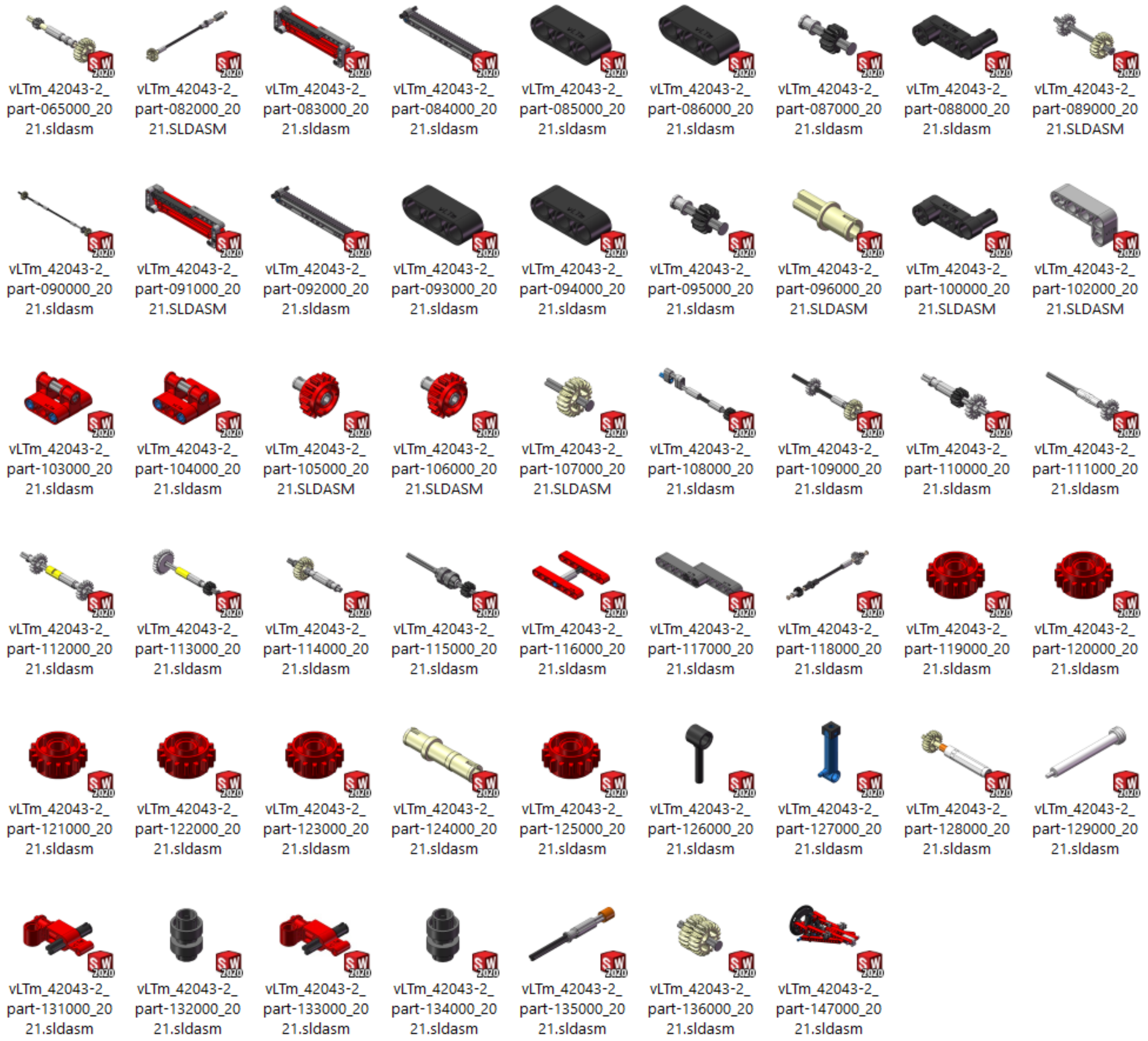


Figure 155. LTM_42043-2 > "G" > All the assemblies.

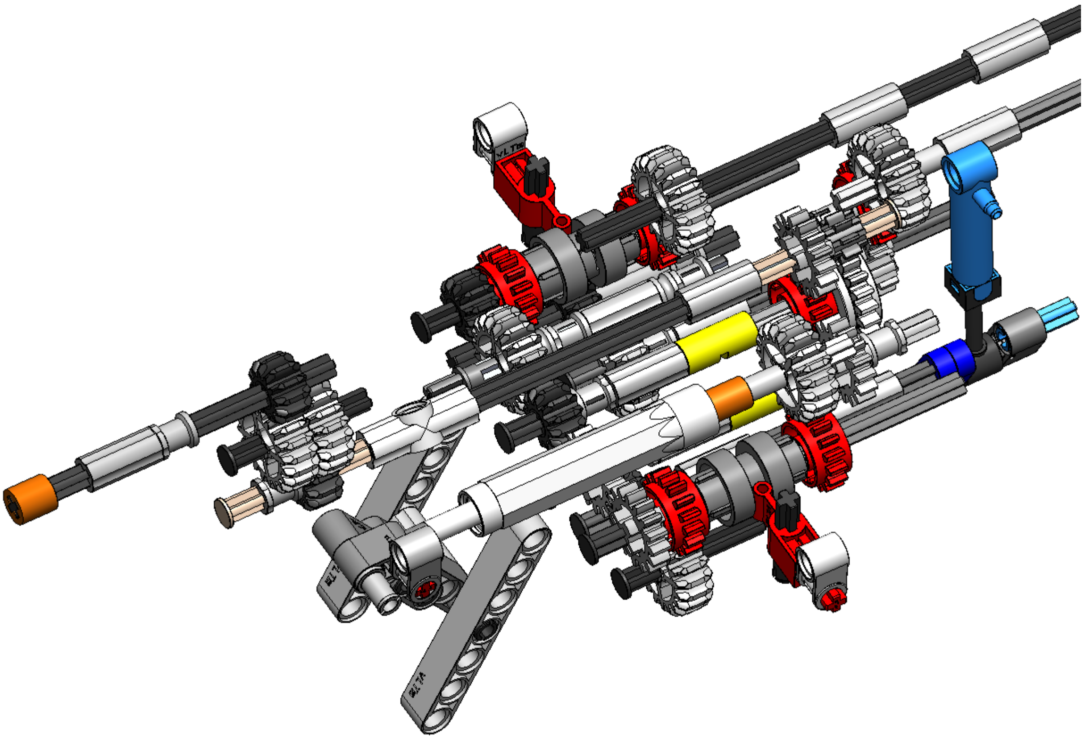


Figure 156. LTM_42043-2 >"G" in SolidWorks® 2020 > A detail of the axis from the engine and how that power can be distributed through gears depending on the wanted option.

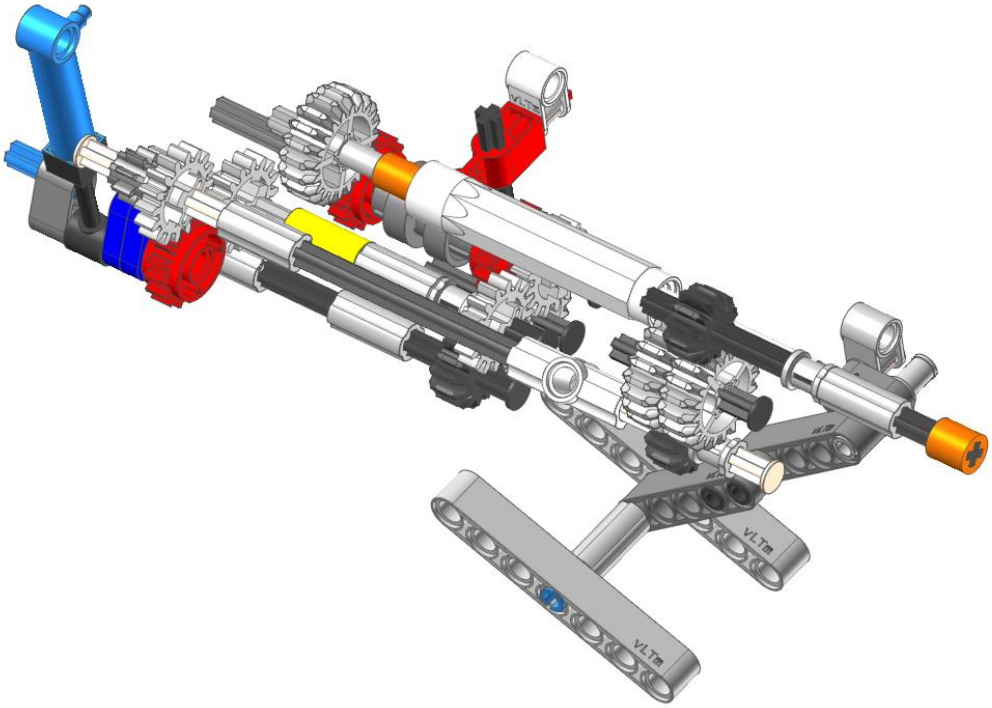


Figure 157. LTM_42043-2 >"G" in SolidWorks® 2020 > The 2 options at the left of the trailer: pump or parking.

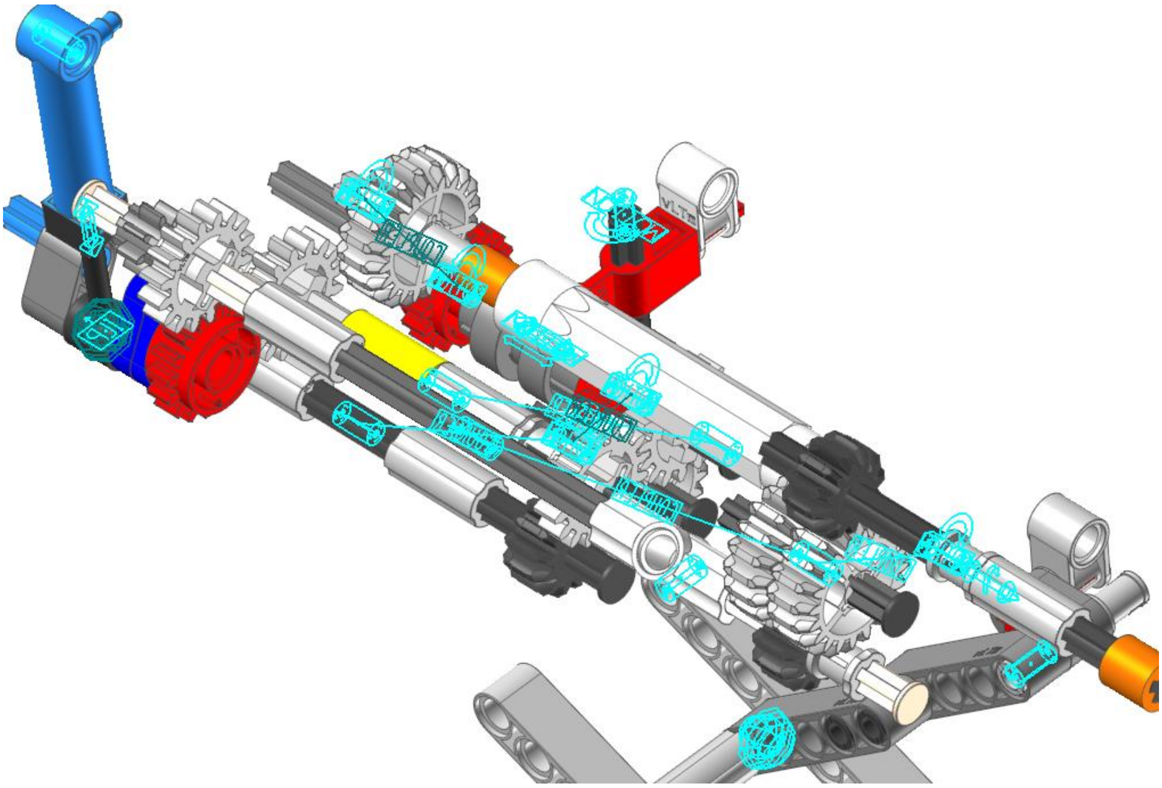


Figure 158. LTM_42043-2 >Self-aligned "G" in Recurdyn > The 2 options at the left: pump or parking.

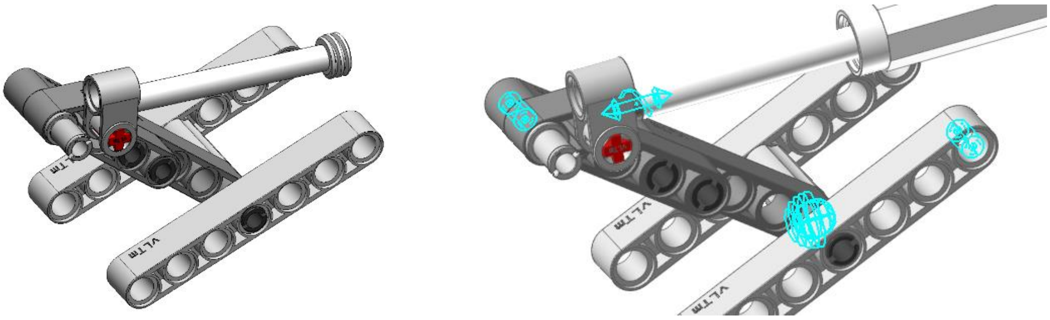


Figure 159. LTM_42043-2 > Self-aligned "G" in SolidWorks® 2020 & Recurdyn > A detail of "parking".

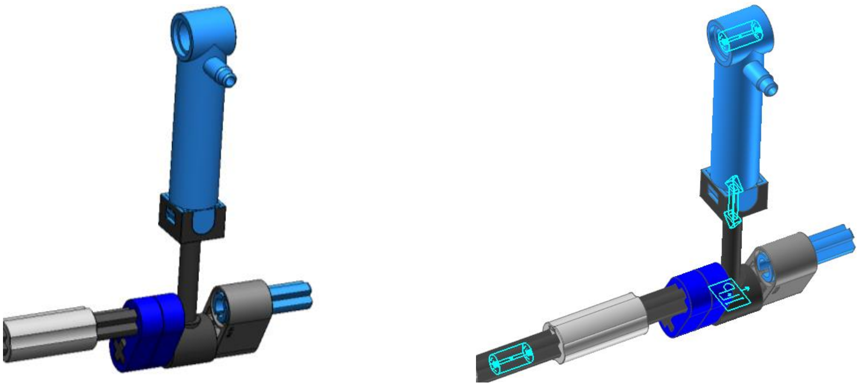


Figure 160. LTM_42043-2 > Self-aligned "G" in SolidWorks® 2020 & Recurdyn > A detail of "pump".

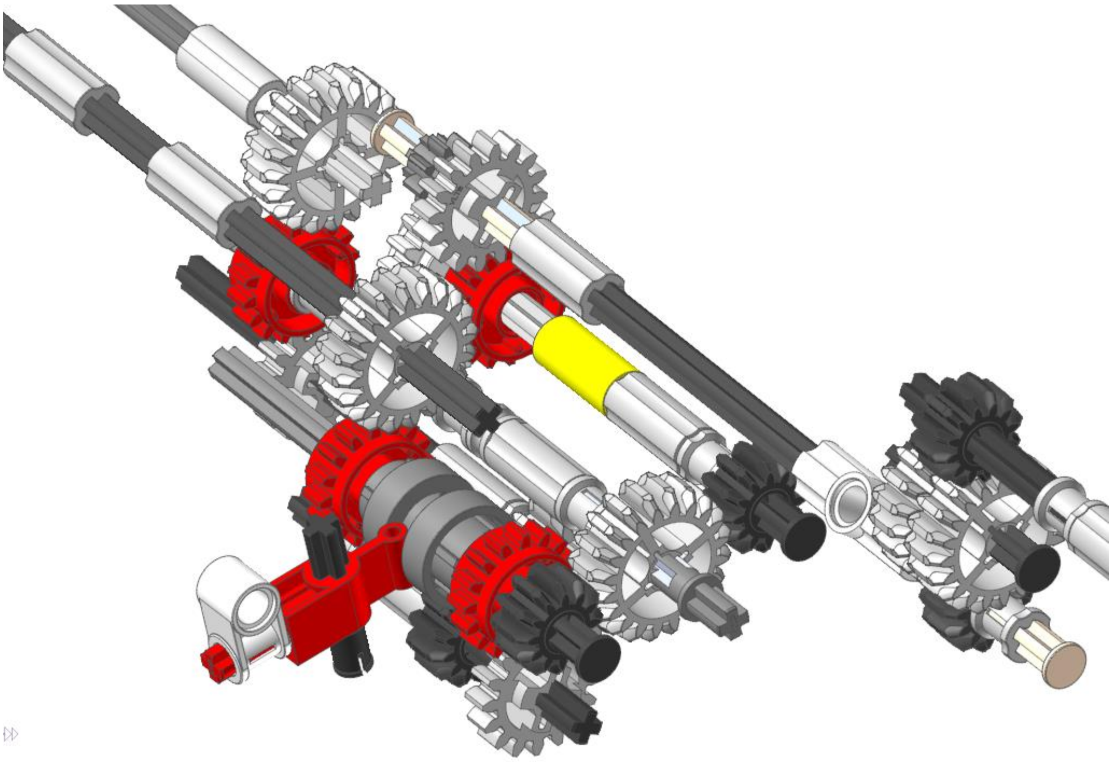


Figure 161. LTM_42043-2 >"G" in SolidWorks® 2020 > The 2 options at the right of the trailer: rotation of the crane or to use the supports.

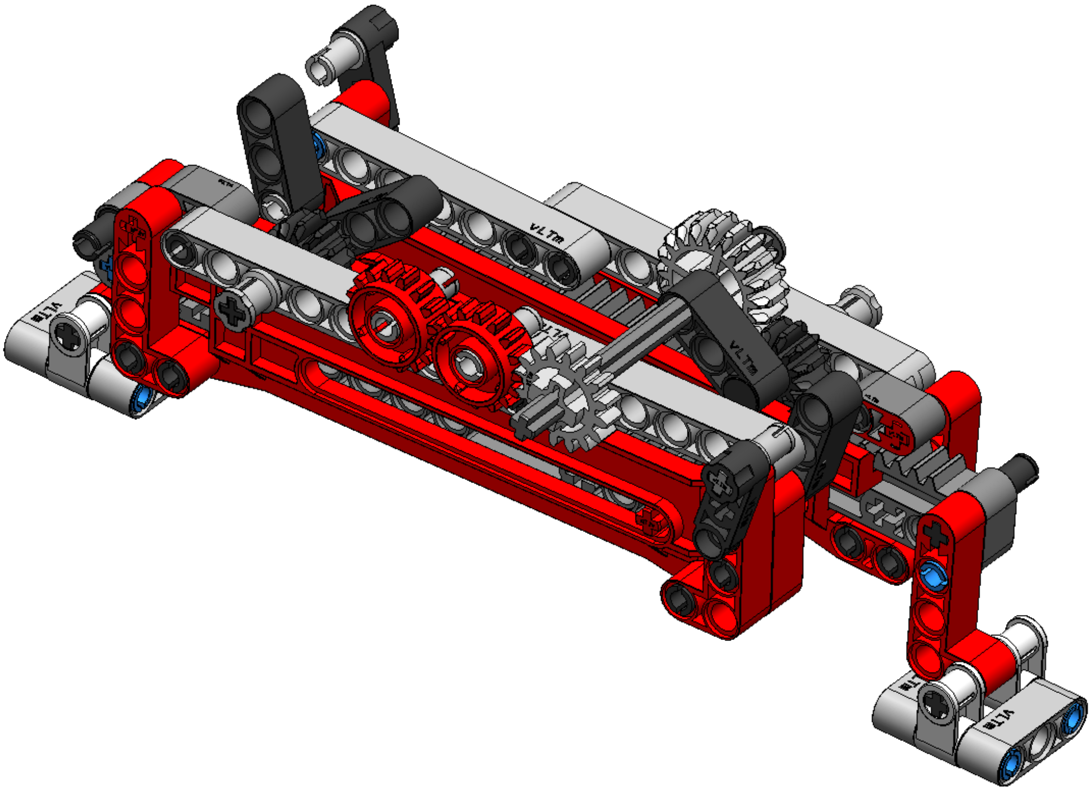


Figure 162. LTM_42043-2 > "G" in SolidWorks® 2020 > A detail of "supports".

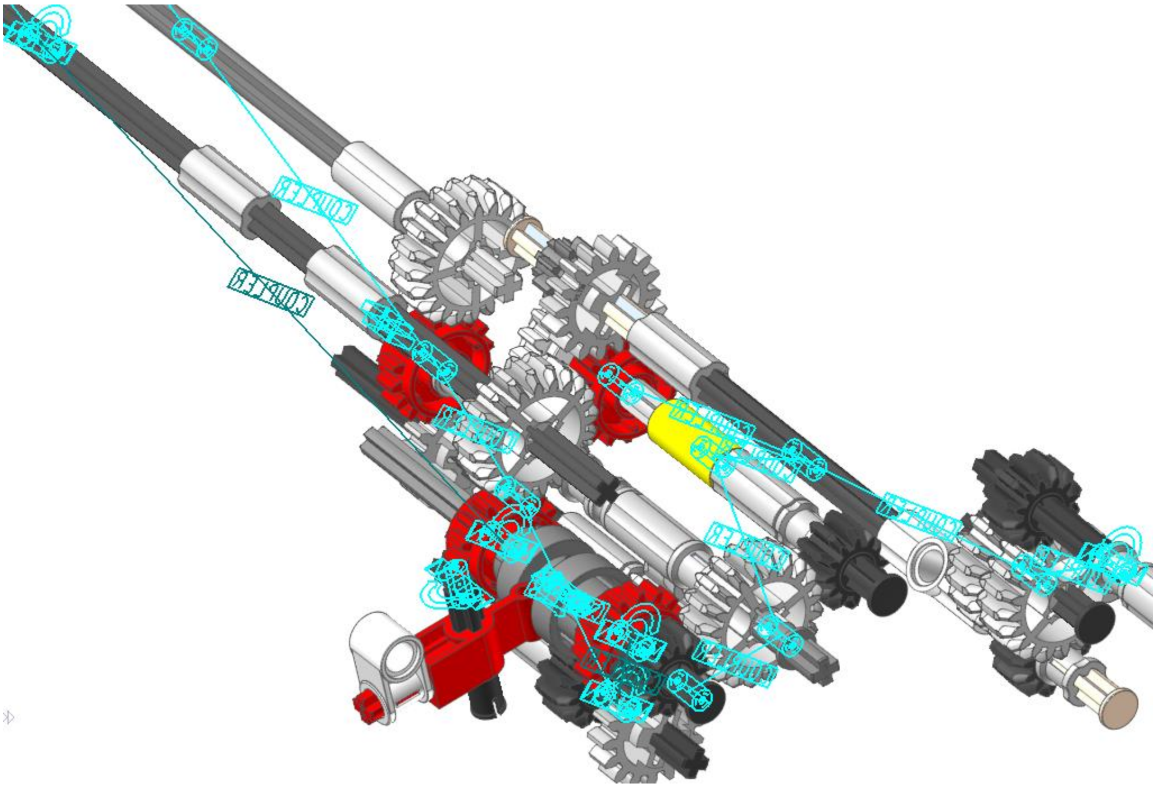


Figure 163. LTM_42043-2 >"G" in Recurdyn > The 2 options at the right: rotation of the crane or to use the supports (1/2).

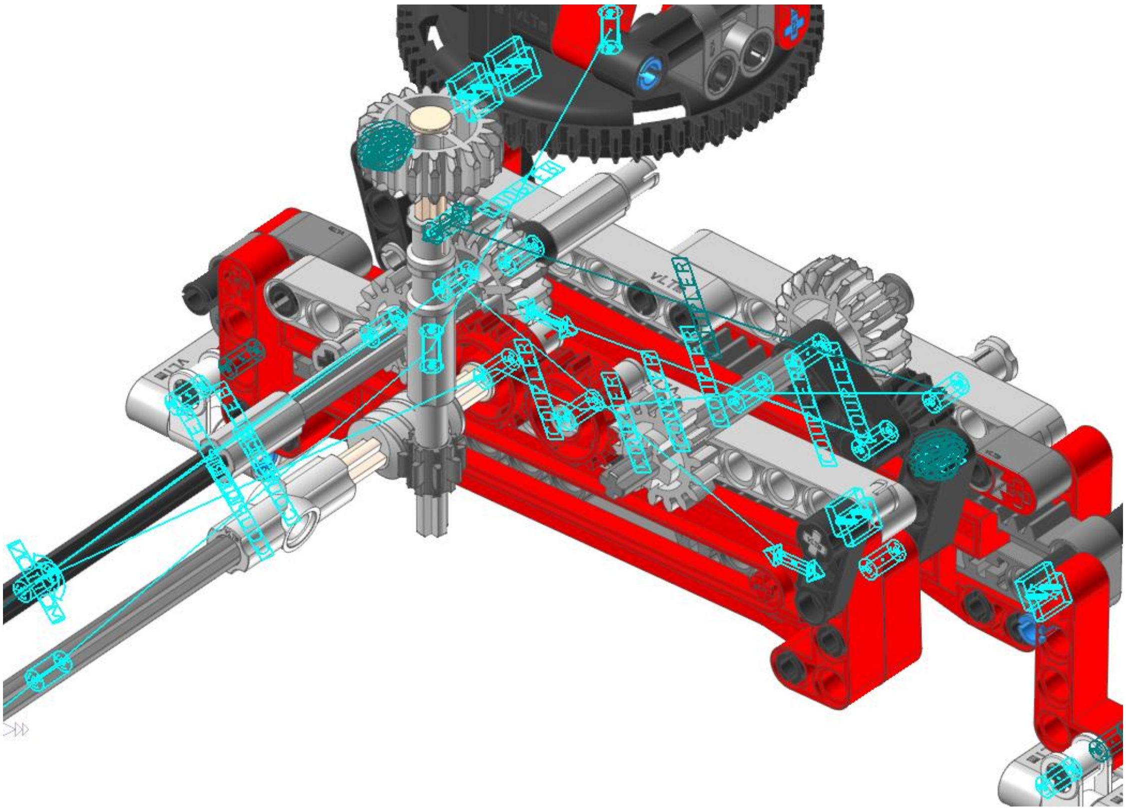


Figure 164. LTM_42043-2 >"G" in Recurdyn > The 2 options at the right: rotation of the crane or to use the supports (2/2).

1.2.2.7. MECHANISM “CRANE”, R

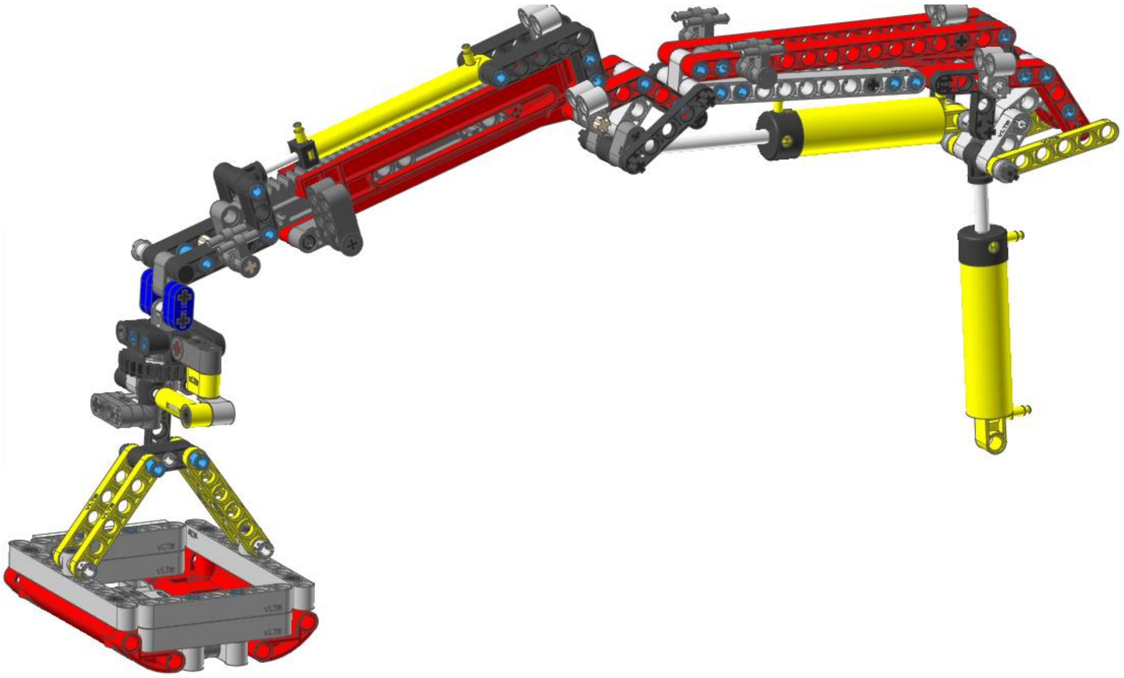


Figure 165. LTM_42043-2 > Mechanism “Crane” (R) in SolidWorks® 2020. FTTG: 147.

vLTm_42043-2_part-146000_2021.SLDASM
vLTm_42043-2_part-148000_2021.SLDASM
vLTm_42043-2_part-149000_2021.sldasm
vLTm_42043-2_part-150000_2021.sldasm
vLTm_42043-2_part-151000_2021.sldasm
vLTm_42043-2_part-152000_2021.sldasm
vLTm_42043-2_part-153000_2021.sldasm
vLTm_42043-2_part-154000_2021.sldasm
vLTm_42043-2_part-155000_2021.sldasm
vLTm_42043-2_part-156000_2021.sldasm
vLTm_42043-2_part-157000_2021.sldasm
vLTm_42043-2_part-158000_2021.sldasm
vLTm_42043-2_part-159000_2021.sldasm
vLTm_42043-2_part-160000_2021.sldasm
vLTm_42043-2_part-161000_2021.sldasm
vLTm_42043-2_part-162000_2021.sldasm
vLTm_42043-2_part-163000_2021.sldasm
vLTm_42043-2_part-164000_2021.sldasm
vLTm_42043-2_part-165000_2021.SLDASM

Figure 166. LTM_42043-2 > “R” > All the assemblies.

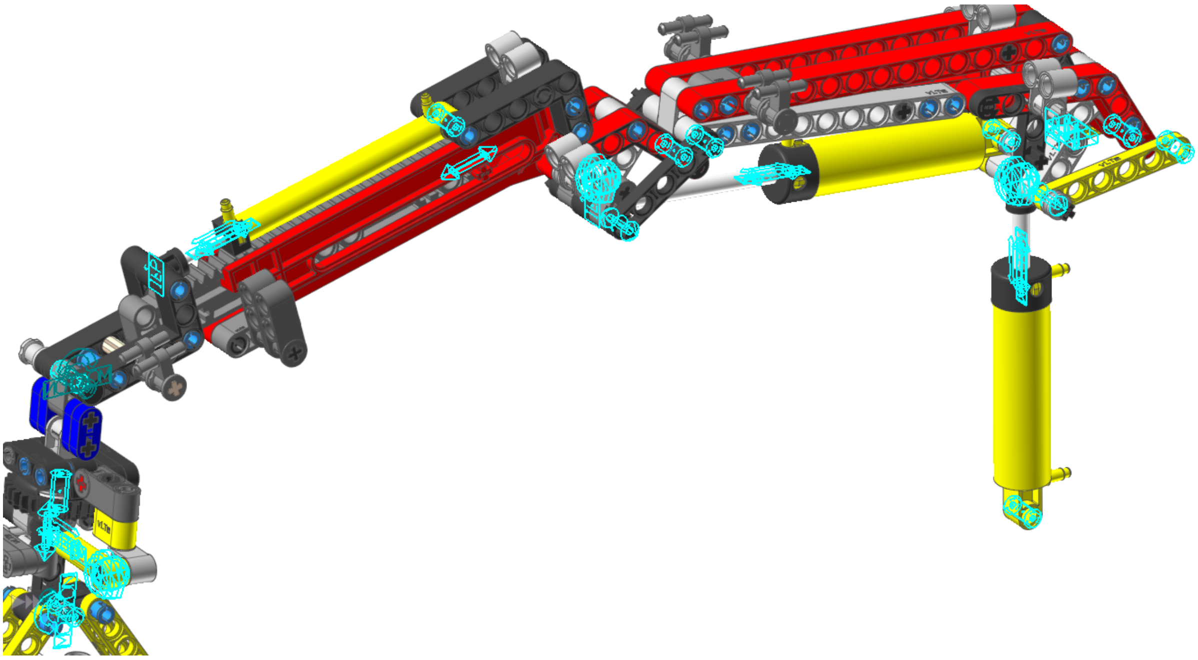
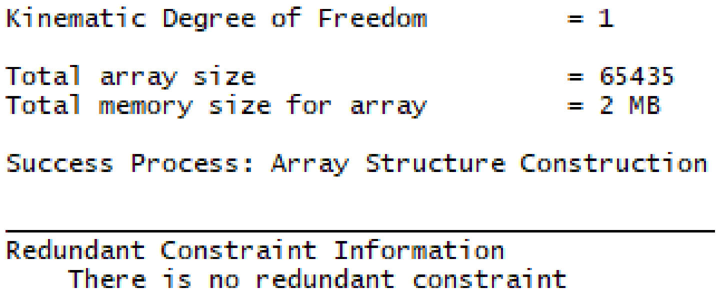


Figure 167. LTM_42043-2 > Self-aligned "R" in Recurdyn.



```
Kinematic Degree of Freedom          = 1

Total array size                     = 65435
Total memory size for array          = 2 MB

Success Process: Array Structure Construction

______________________________________________
Redundant Constraint Information
    There is no redundant constraint
```

Figure 168. LTM_42043-2 > "R" has 5 degrees of freedom: 4 pneumatic cylinders to be controlled and 1 basket with free rotation. The rotation of the whole crane is controlled in the mechanism "G".

1.2.2.7.1. MECHANISM “CRANE”> DOWN, RD

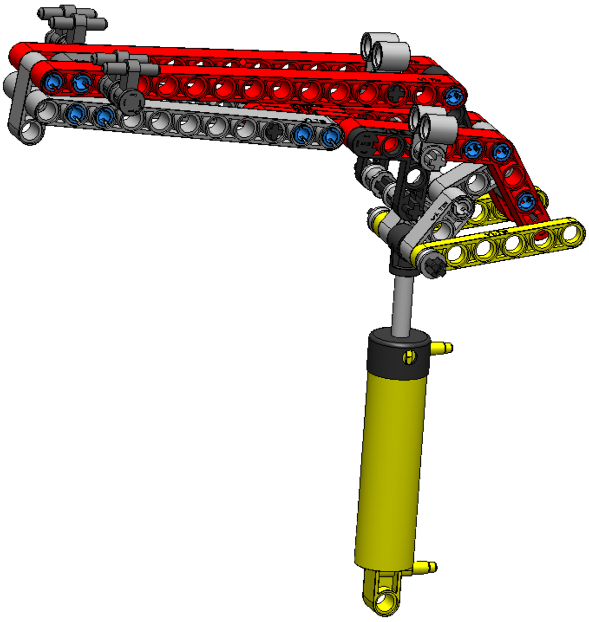


Figure 169. LTM_42043-2 > “R” > “RD” in SolidWorks® 2020. FTTG: 147.

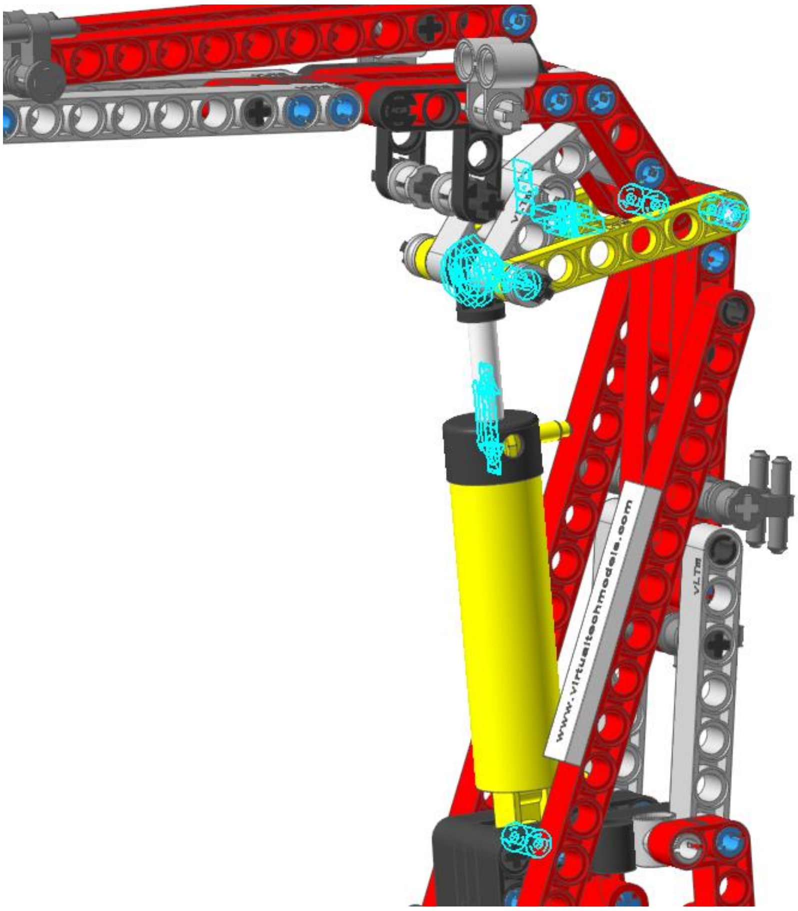


Figure 170. LTM_42043-2 > Self-aligned “R” > “RD” in Recurdyn. Its FTTG piece is shown.

1.2.2.7.2. MECHANISM “CRANE”> MIDDLE, RM

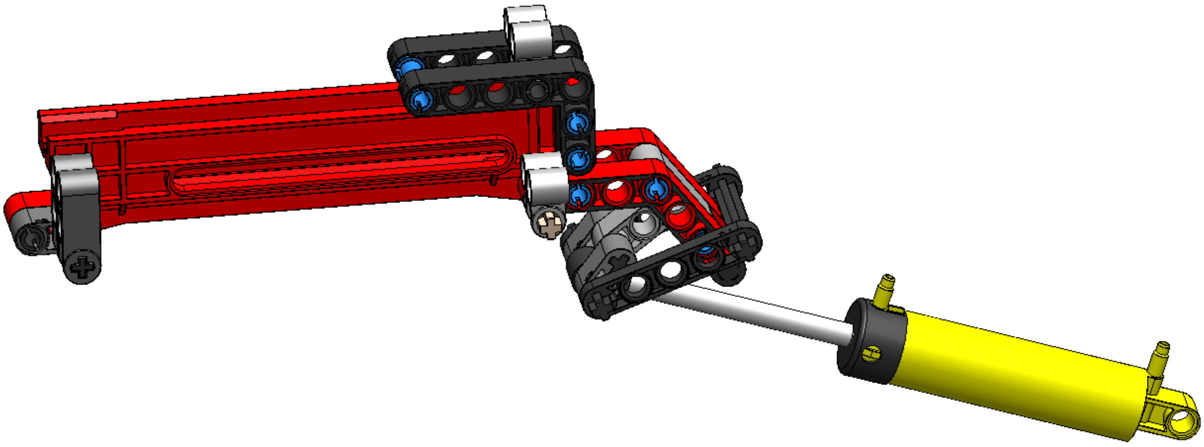


Figure 171. LTM_42043-2 > “R” > “RM” in SolidWorks® 2020. FTTG: 149.

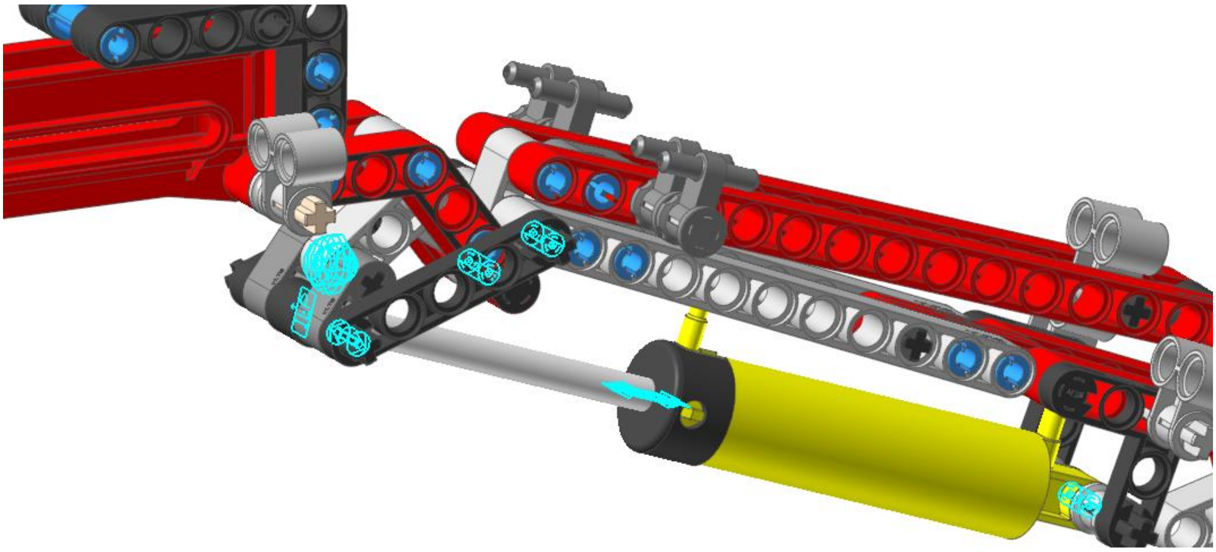


Figure 172. LTM_42043-2 > Self-aligned “R” > “RM” in Recurdyn. Its FTTG piece is shown.

1.2.2.7.3. MECHANISM “CRANE” > UP, RU

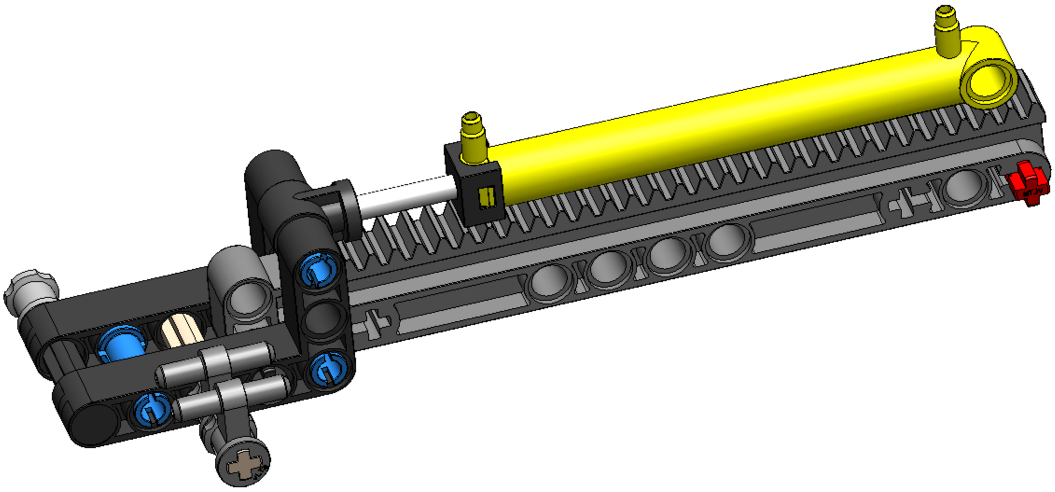


Figure 173. LTM_42043-2 > “R” > “RU” in SolidWorks® 2020. FTTG: 157.

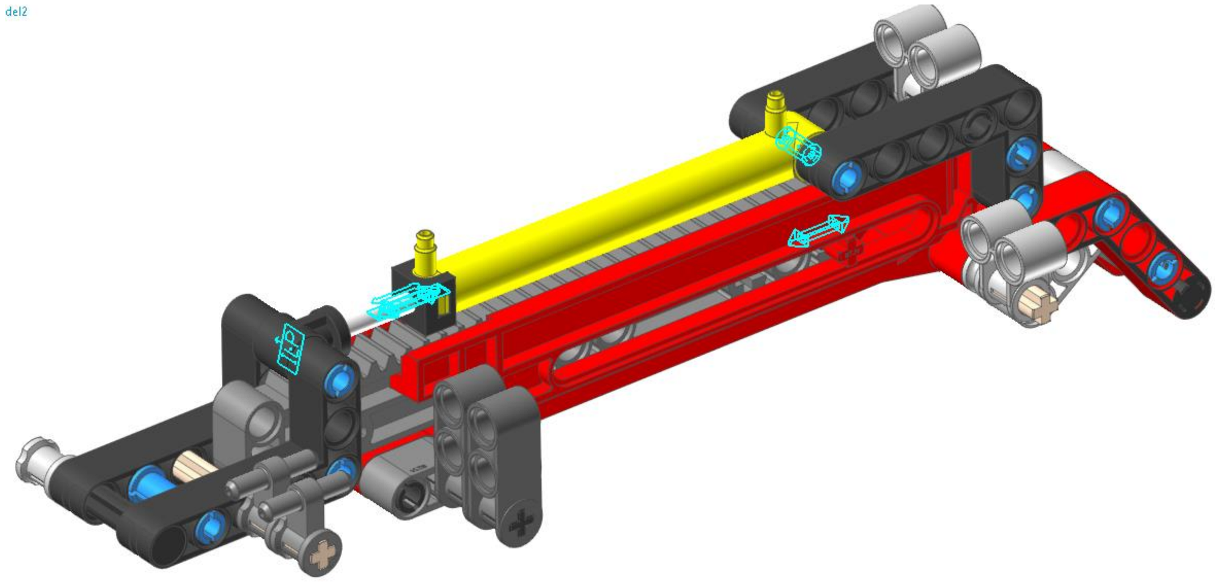


Figure 174. LTM_42043-2 > Self-aligned “R” > “RU” in Recurdyn. Its FTTG piece is shown.

1.2.2.7.4. MECHANISM “CRANE” > BASKET, RB

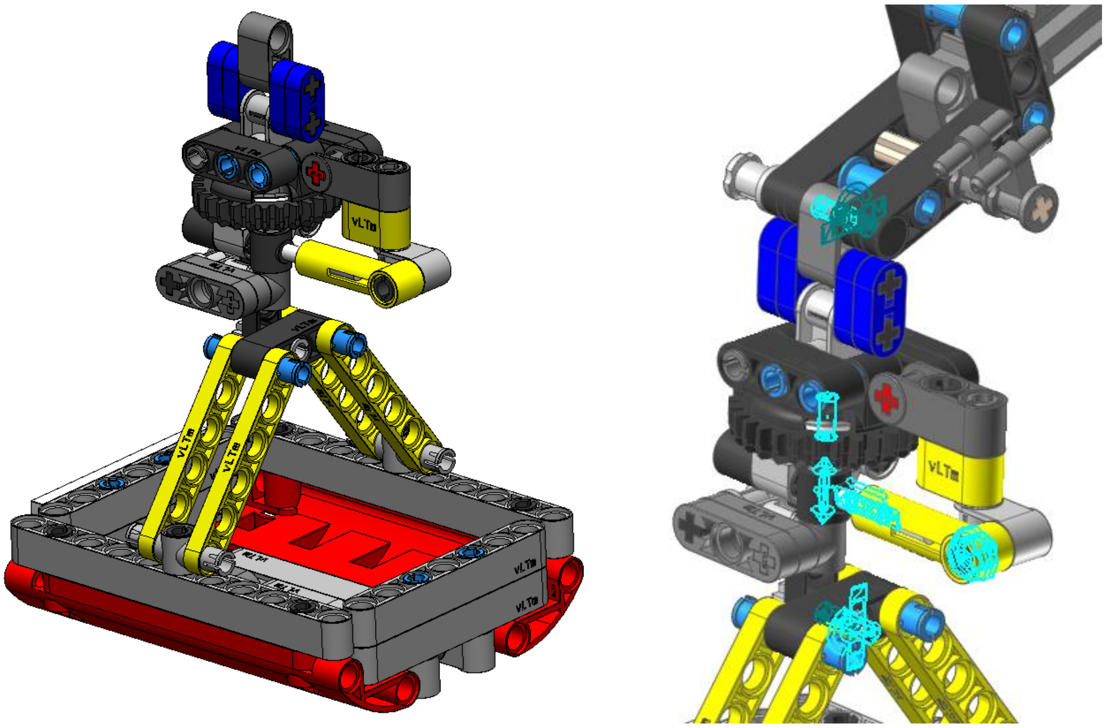


Figure 175. LTM_42043-2 > “R” > “RB” in SolidWorks® 2020 (left; FTTG: 158) and Self-aligned “RB” in Recurdyn (right; its FTTG piece is shown).

1.2.3. SIMULATION: VLTM_42043-2

vLTm_42043-2_2021
-wB-16-motion-a w_

vLTm_42043-2_2021
-wB-16-motion-a w_

vLTm_42043-2_2021
-wB-16-motion-a w_

vLTm_42043-2_2021
-wB-16-motion-a w_

vLTm_42043-2_2021
-wB-16-motion-a w_

vLTm_42043-2_2021
-wB-16-motion-a w_

1.3. MODEL #1: LTM_8862

It is a **backhoe**.

1.3.1. ASSEMBLY: LTM_8862

Figure 176. LTM_8862 in real life. Source: this is the source (enstock3w.top/).

Figure 177. LTM_8862 in COSMOSMotion 2007.

Figure 178. LTM_8862 > A real life machine (1/4). Source: this is the source (bossmachinery.nl).

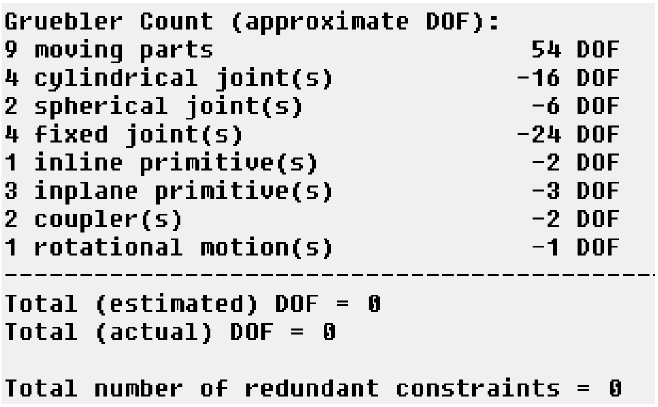
Figure 179. LTM_8862 > A real life machine (2/4). Source: this is the source (bossmachinery.nl).

Figure 180. LTM_8862 > A real life machine (3/4). Source: this is the source (bossmachinery.nl).

Figure 181. LTM_8862 > A real life machine (4/4). Source: this is the source (bossmachinery.nl).

1.3.2. MECHANISMS: VLTM_8862

Figure 182 LTM_8862 > Self-aligned Mechanism "Direction" (D) in SolidWorks® 2007 (left) and in COSMOSMotion 2007 (right). 1 DOF to be controlled to move the vehicle to the left or to the right. FTTG: 001.



```
Gruebler Count (approximate DOF):
9 moving parts                     54 DOF
4 cylindrical joint(s)            -16 DOF
2 spherical joint(s)               -6 DOF
4 fixed joint(s)                  -24 DOF
1 inline primitive(s)              -2 DOF
3 inplane primitive(s)             -3 DOF
2 coupler(s)                       -2 DOF
1 rotational motion(s)             -1 DOF
-------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 183. LTm_42124-1 > Self-aligned "D" > COSMOSMotion 2007 Simulation Panel.

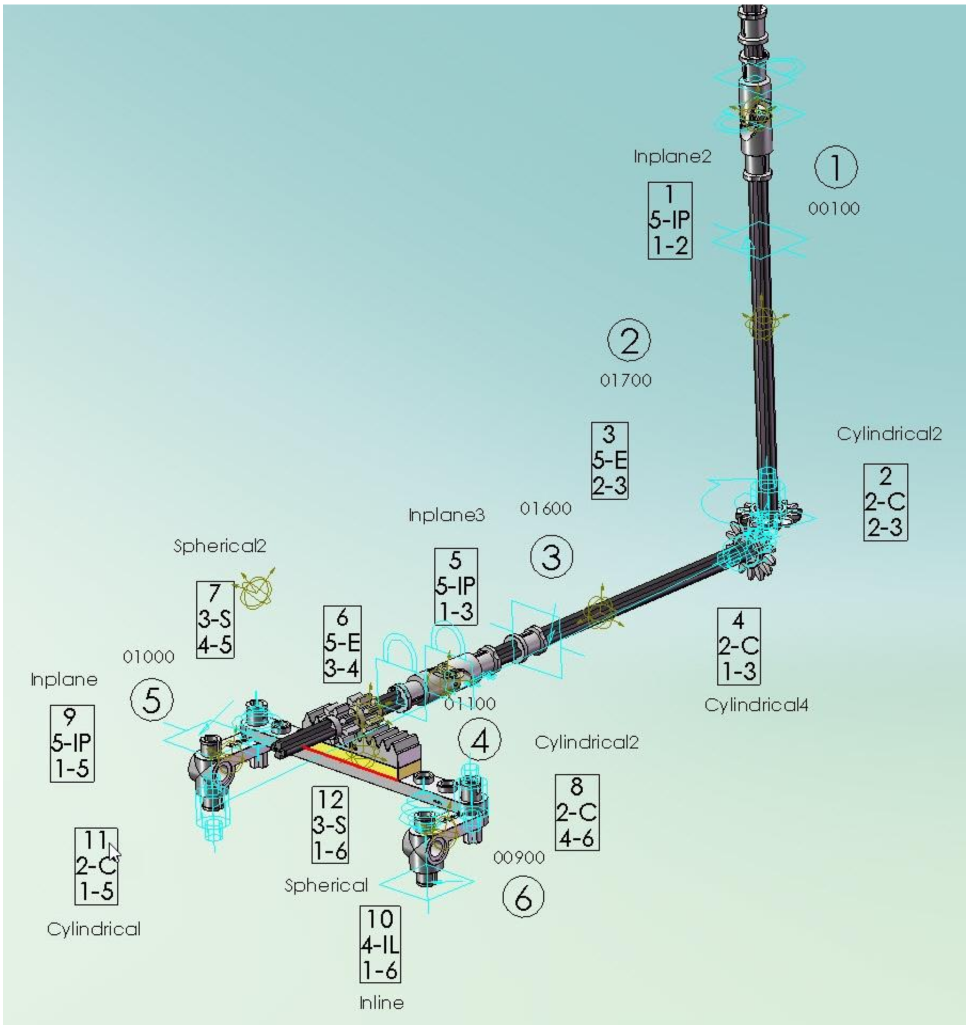


Figure 184. LTM_8862 > "D" > Joints in SolidWorks® 2007.

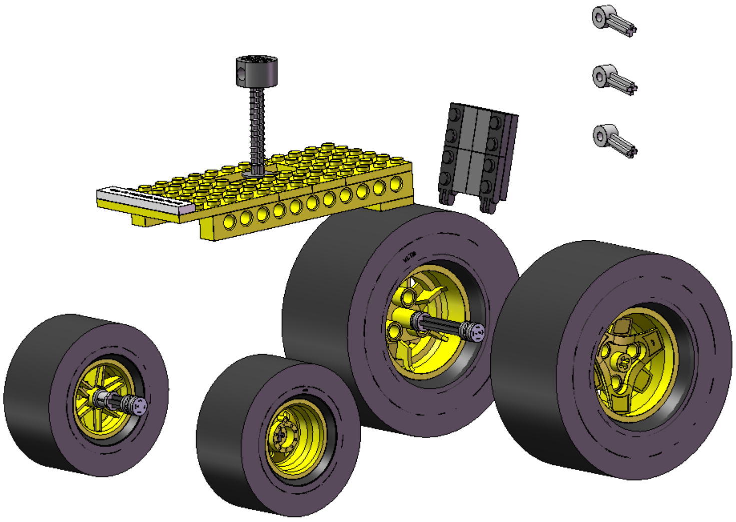
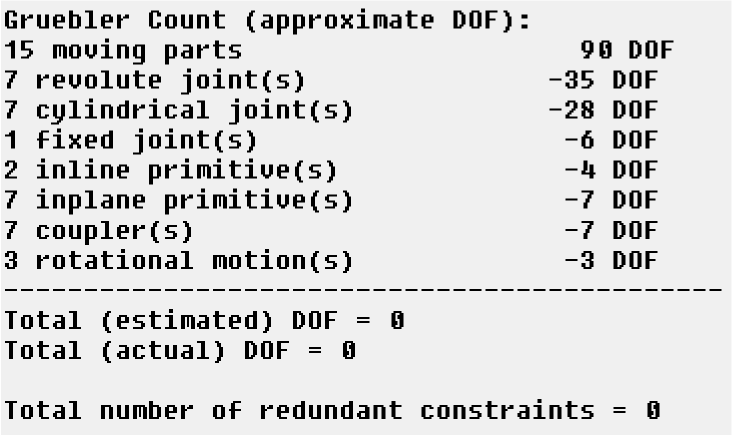


Figure 185. LTM_8862 > Mechanism "All opened kinematic chains" (A) in SolidWorks® 2007. These pieces are not needed to close a kinematic chain. They are constrained by a R joint. FTTG: 001.

Figure 186. LTM_8862 > Mechanism "Support" (U) in SolidWorks® 2007. FTTG: 001.

Figure 187. LTM_8862 > Self-aligned "U" in COSMOSMotion 2007. 1 DOF to be controlled: to extend or to contact the supports.



```
Gruebler Count (approximate DOF):
15 moving parts                      90 DOF
7 revolute joint(s)                 -35 DOF
7 cylindrical joint(s)              -28 DOF
1 fixed joint(s)                     -6 DOF
2 inline primitive(s)                -4 DOF
7 inplane primitive(s)               -7 DOF
7 coupler(s)                         -7 DOF
3 rotational motion(s)               -3 DOF
---------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 188. LTm_42124-1 > Self-aligned "U" > COSMOSMotion 2007 Simulation Panel.

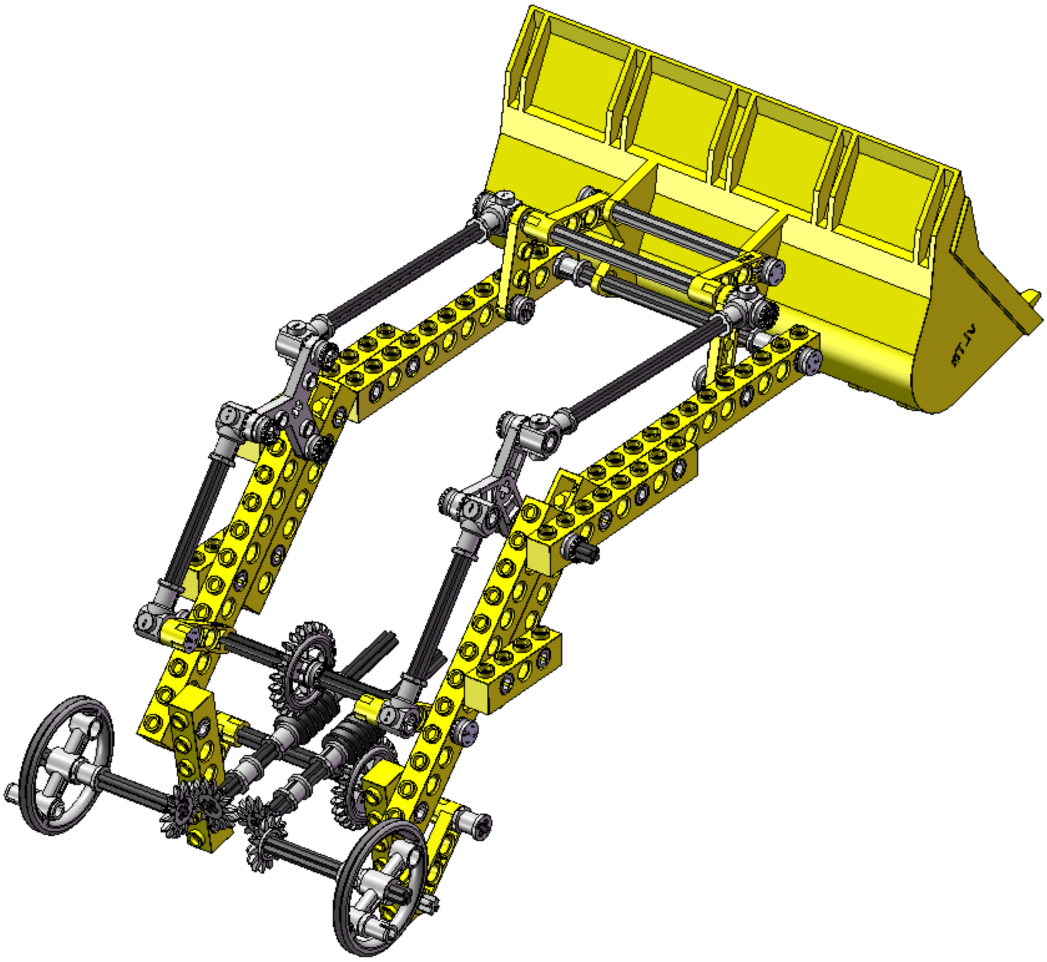


Figure 189. LTM_8862 > Mechanism "Frontal shovel" (FS) in SolidWorks® 2007. FTTG: 001.

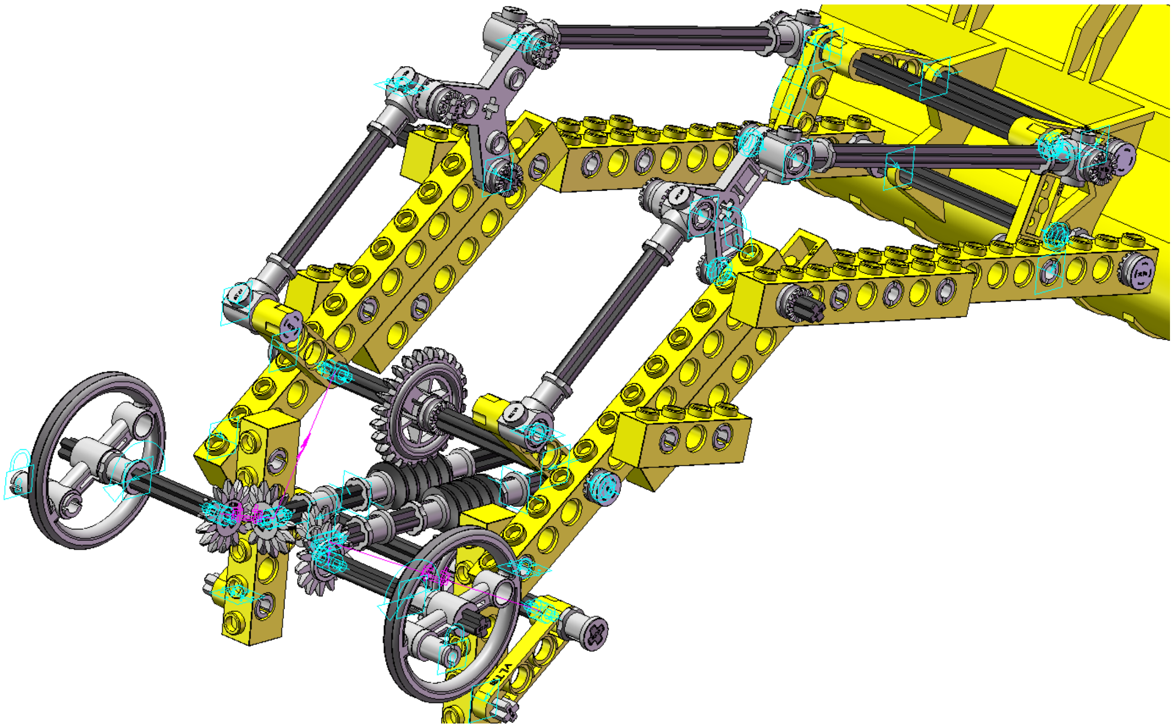
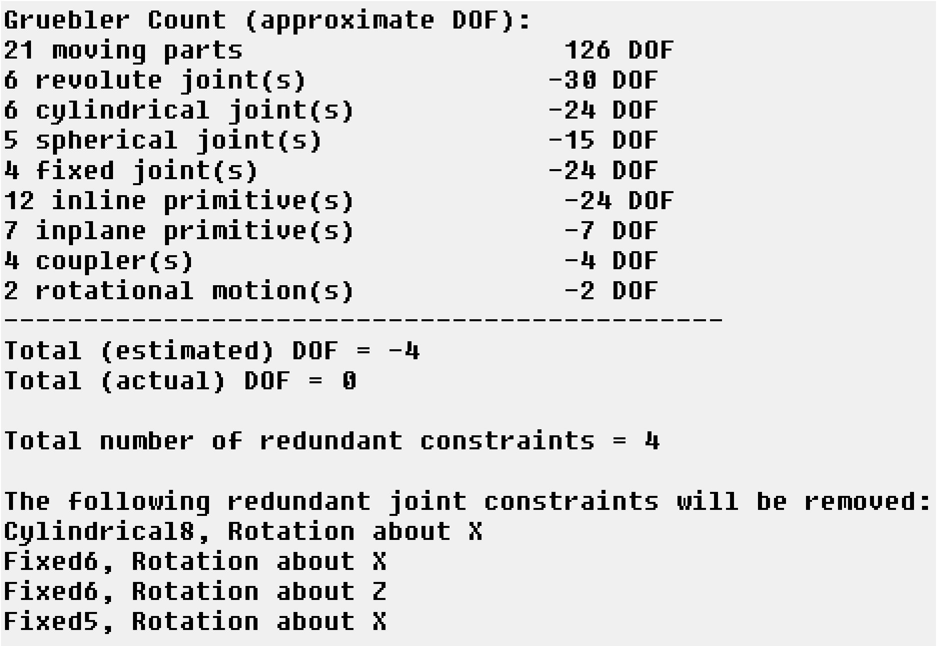


Figure 190. LTM_8862 > Self-aligned "FS" in COSMOSMotion 2007. 2 DOF to be controlled: raising (up or down) and rotating (download or load) the shovel.



```
Gruebler Count (approximate DOF):
21 moving parts                    126 DOF
6 revolute joint(s)               -30 DOF
6 cylindrical joint(s)            -24 DOF
5 spherical joint(s)              -15 DOF
4 fixed joint(s)                  -24 DOF
12 inline primitive(s)             -24 DOF
7 inplane primitive(s)             -7 DOF
4 coupler(s)                       -4 DOF
2 rotational motion(s)             -2 DOF
---------------------------------------------
Total (estimated) DOF = -4
Total (actual) DOF = 0

Total number of redundant constraints = 4

The following redundant joint constraints will be removed:
Cylindrical8, Rotation about X
Fixed6, Rotation about X
Fixed6, Rotation about Z
Fixed5, Rotation about X
```

*Figure 191. LTm_42124-1 > **NO** Self-aligned "FS" > COSMOSMotion 2007 Simulation Panel.*

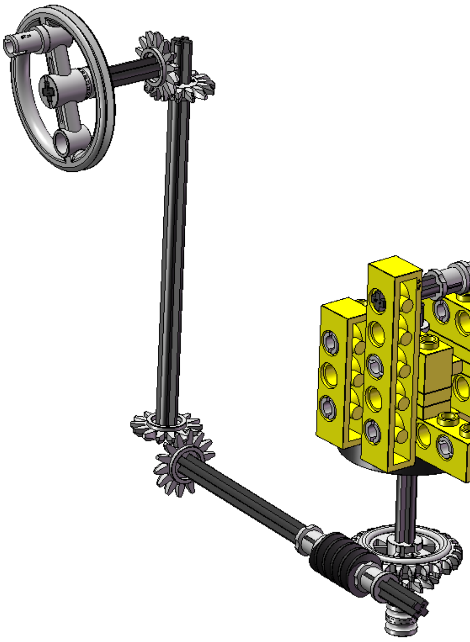


Figure 192. LTM_8862 > Mechanism "Rotation of back shovel" (RBS) in SolidWorks® 2007. 1 DOF: rotation of the chassis of the back shovel

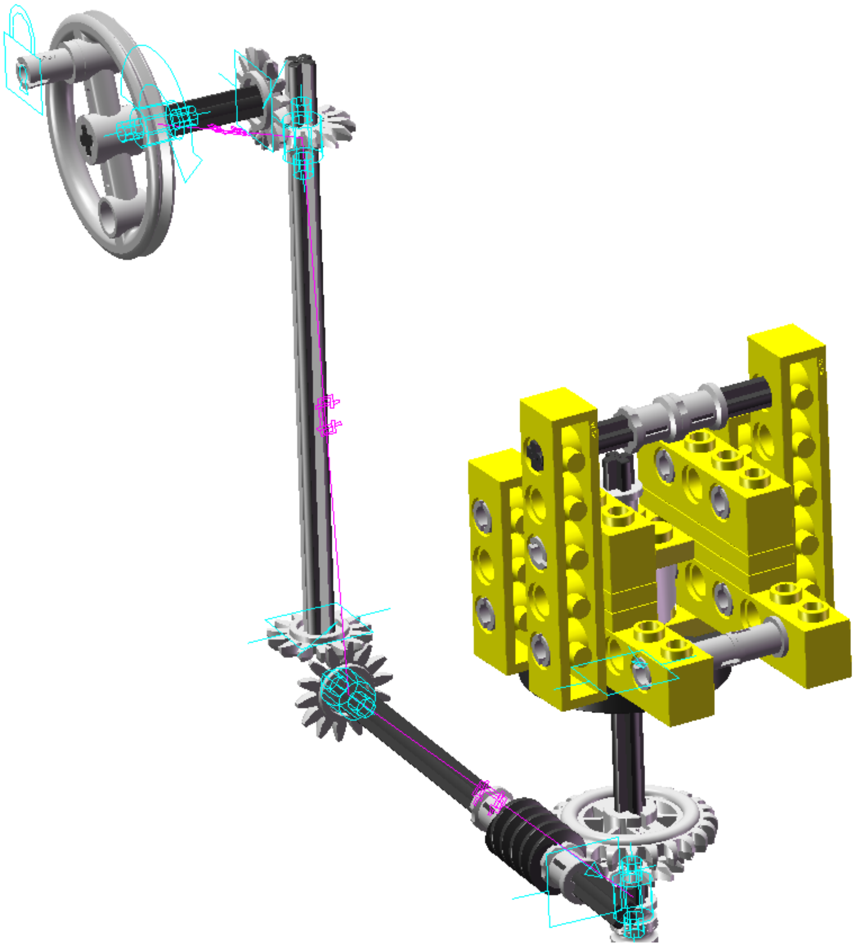
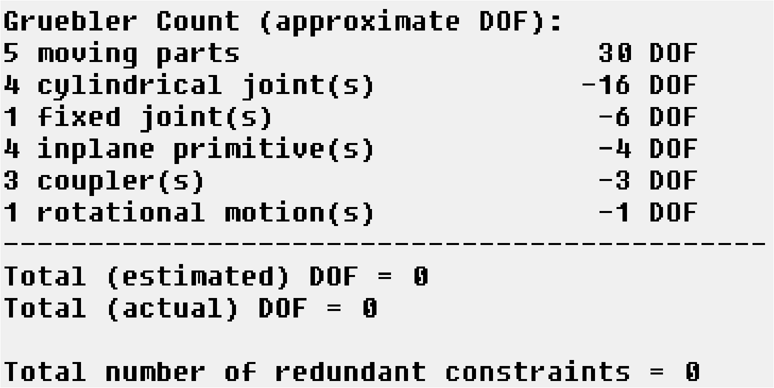


Figure 193. LTM_8862 > Self-aligned "RBS" in COSMOSMotion 2007. 1 DOF: rotation of the chassis of the back shovel.



```
Gruebler Count (approximate DOF):
5 moving parts                        30 DOF
4 cylindrical joint(s)               -16 DOF
1 fixed joint(s)                      -6 DOF
4 inplane primitive(s)                -4 DOF
3 coupler(s)                          -3 DOF
1 rotational motion(s)                -1 DOF
---------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 194. LTm_42124-1 > Self-aligned “RBS” > COSMOSMotion 2007 Simulation Panel.

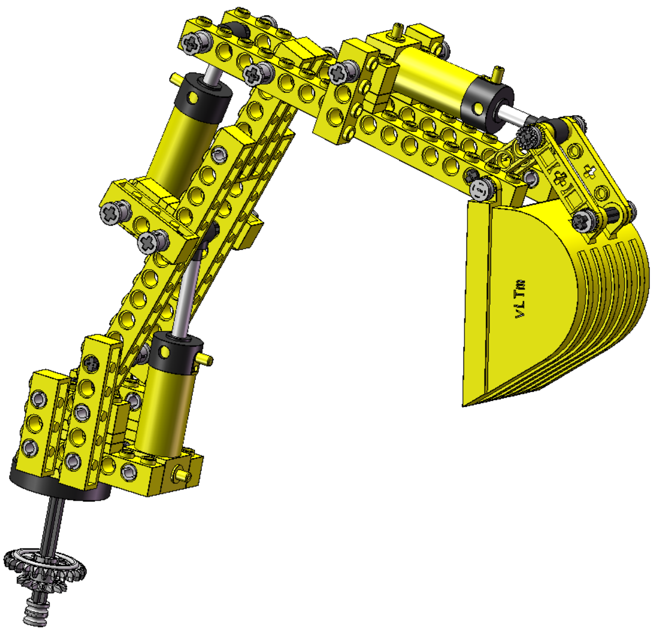
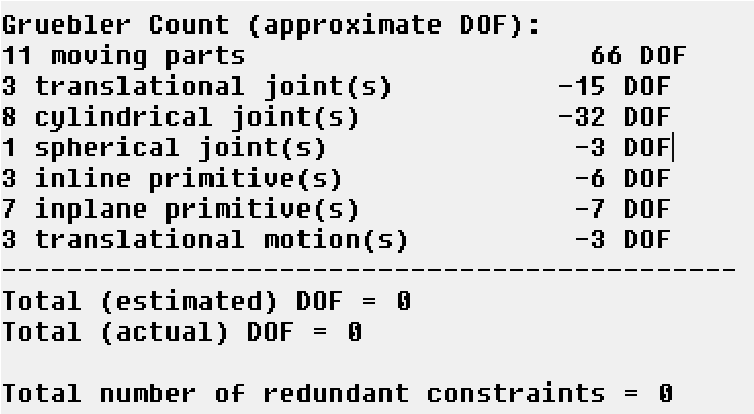


Figure 195. LTM_8862 > Mechanism “Control of the back shovel” (CBS) in SolidWorks® 2007.



```
Gruebler Count (approximate DOF):
11 moving parts                       66 DOF
3 translational joint(s)             -15 DOF
8 cylindrical joint(s)               -32 DOF
1 spherical joint(s)                  -3 DOF
3 inline primitive(s)                 -6 DOF
7 inplane primitive(s)                -7 DOF
3 translational motion(s)             -3 DOF
---------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 196. LTm_42124-1 > Self-aligned “CBS” > COSMOSMotion 2007 Simulation Panel.

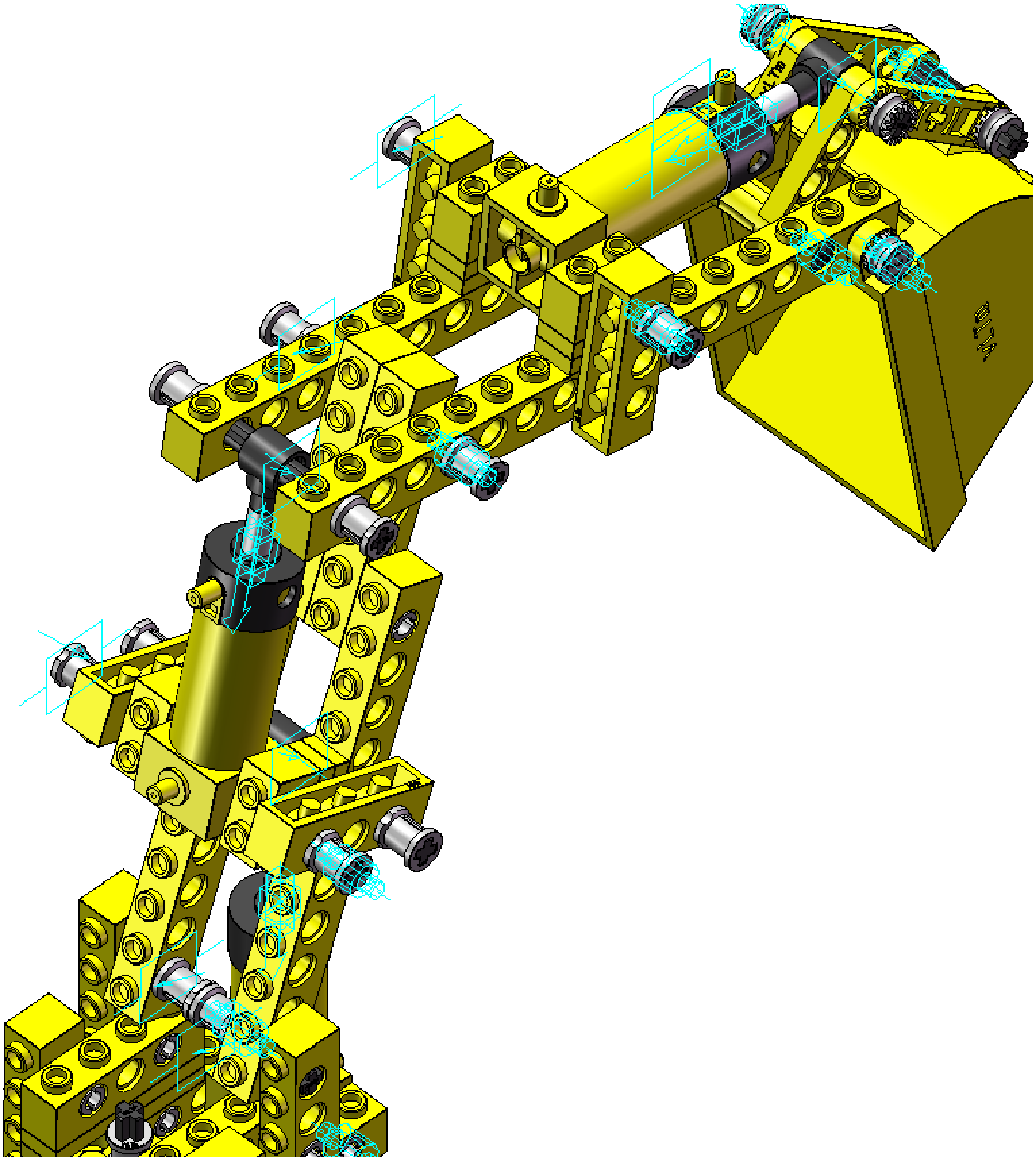
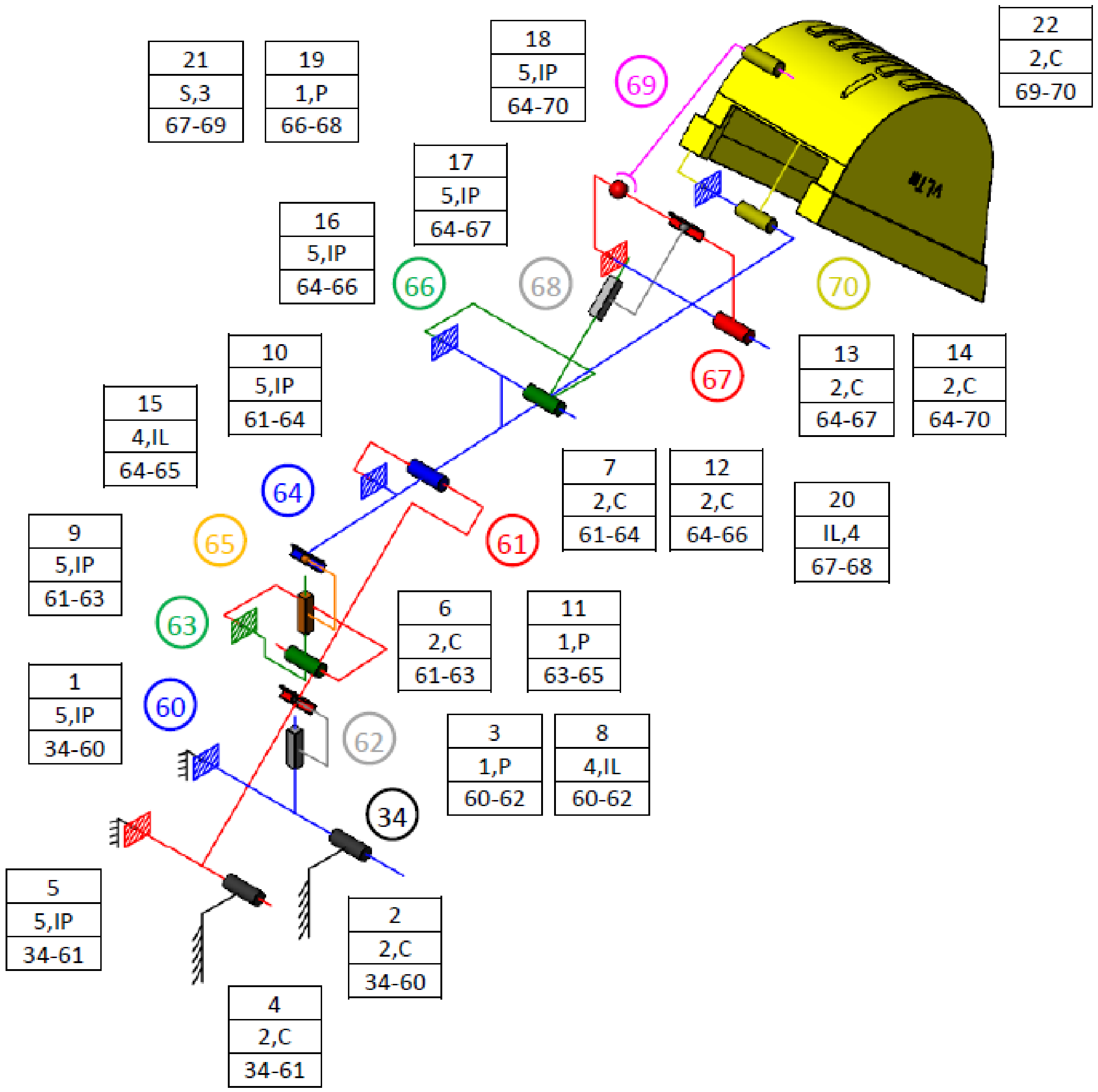


Figure 197. LTM_8862 > Self-aligned "CBS" in COSMOSMotion 2007. 3 DOF to be controlled, one per each pneumatic cylinder. The rotation of the shovel is controlled by mechanism "rotation of back wheel".



$$M_{3D} = 6 * (12 - 22 - 1) + 3 + 2x8 + 7x5 + 3 + 3x4 = 3$$

$$N.R.E. = M_a - M_{3D} = 3 - 3 = 0$$

$$\boldsymbol{M = 3\ GDL}$$

Figure 198. LTM_8862 > “CBS” > Spatial kinematic diagram.

1.3.3. SIMULATION: VLTM_8862

vLTm_8862-1_2015-
motion.mp4

vLTm_8862-1_2015-
motion-base.avi

vLTm_8862-1-mcv-b
ase_2021-a-1920x10

1.4. MODEL #2: LTM_8047-1

It is a **crawler excavator**.

1.4.1. ASSEMBLY: LTM_8047-1

Figure 199. LTM_8047-1 in real life. Source: this is the source

Figure 200. LTM_8047-1 in SolidWorks® 2020 with original chains (left), and after creating chains with the “Track_LM” toolkit in Recurdyn (right).

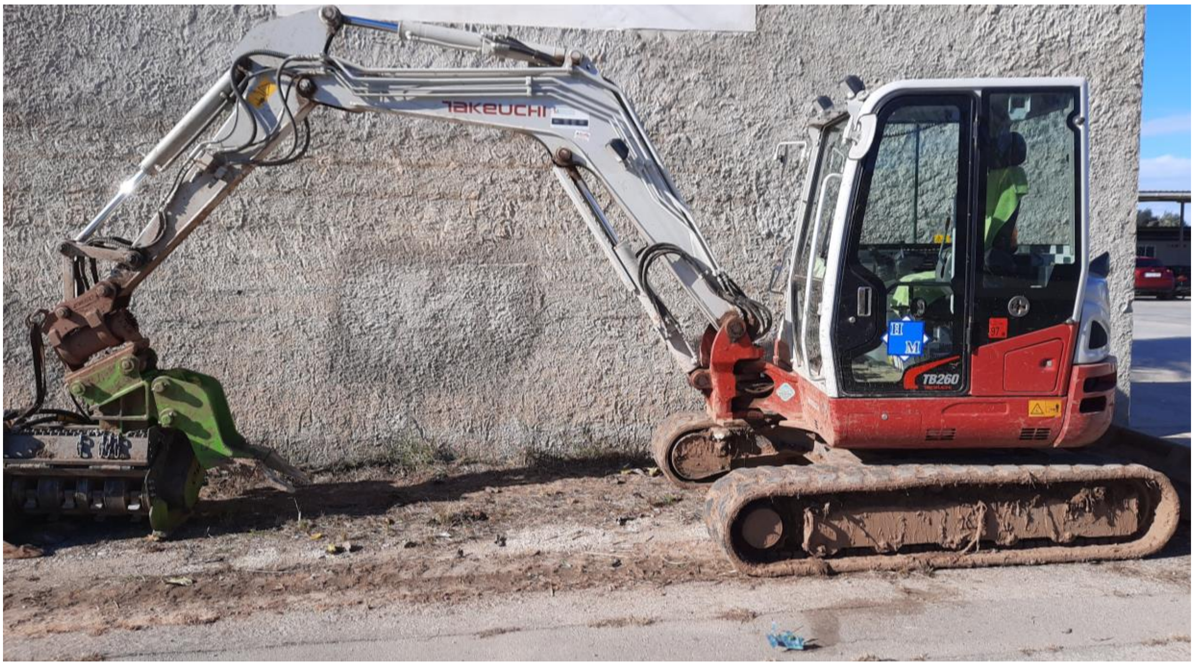


Figure 201. LTM_8047-1 > A real life machine (1/11). > All.

Figure 202. LTM_8047-1 > A real life machine (2/11). > All. Source: this is the source (bossmachinery.nl).

1.4.2. MECHANISMS: VLTM_8047-1

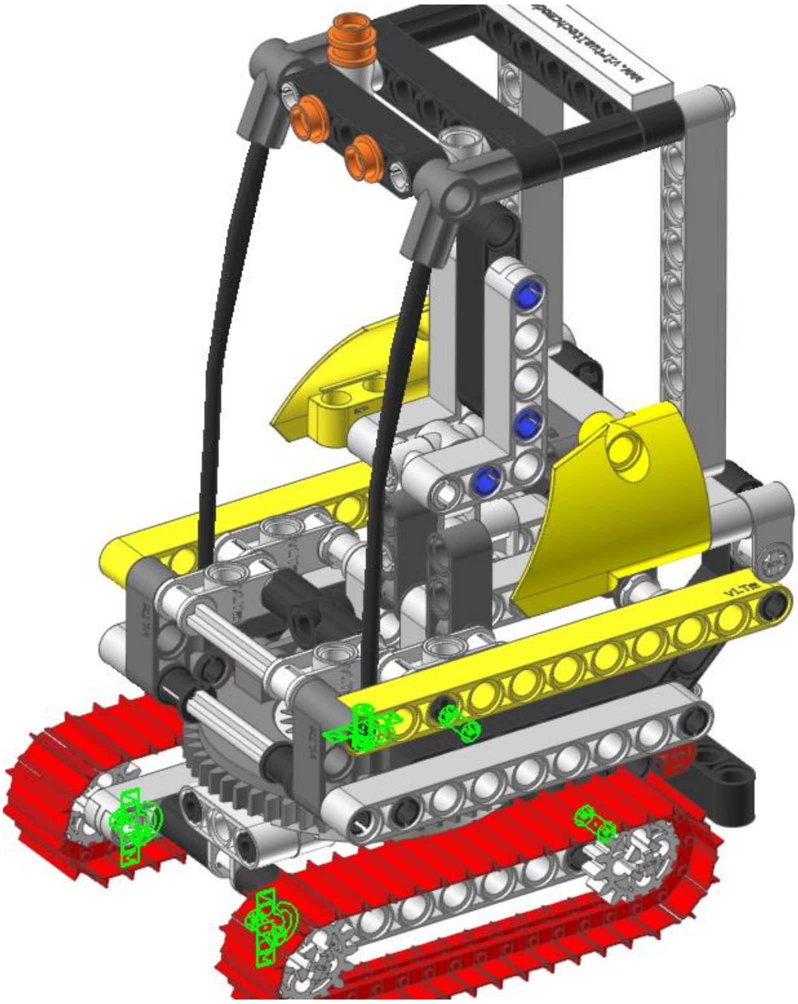


Figure 203. LTM_8047-1 > Self-aligned Mechanism "Driving chains and chassis' rotation" in Recurdyn. 3 DOF to be controlled: 2 driving chains and rotation of the top chassis with respect to the bottom chassis.

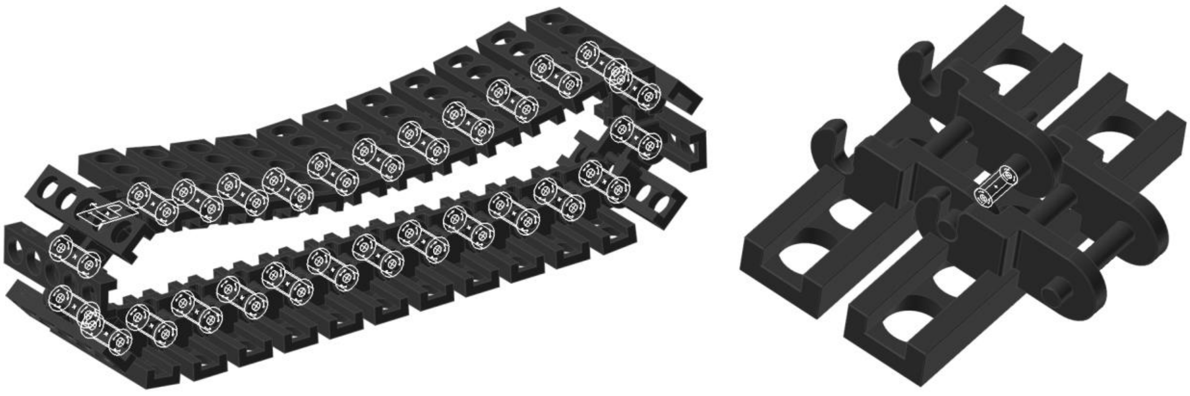


Figure 204. LTM_8047-1 > How to self-align a chain: all R joints but one IL (1/2).

Figure 205. LTM_8047-1 > A real life machine (3/11). > Rotation of shovel with respect to cabin's chassis, rotation of cabin's chassis with respect to chains' chassis, and chains.

Figure 206. LTM_8047-1 > A real life machine (4/11). > Rotation of shovel with respect to cabin's chassis.

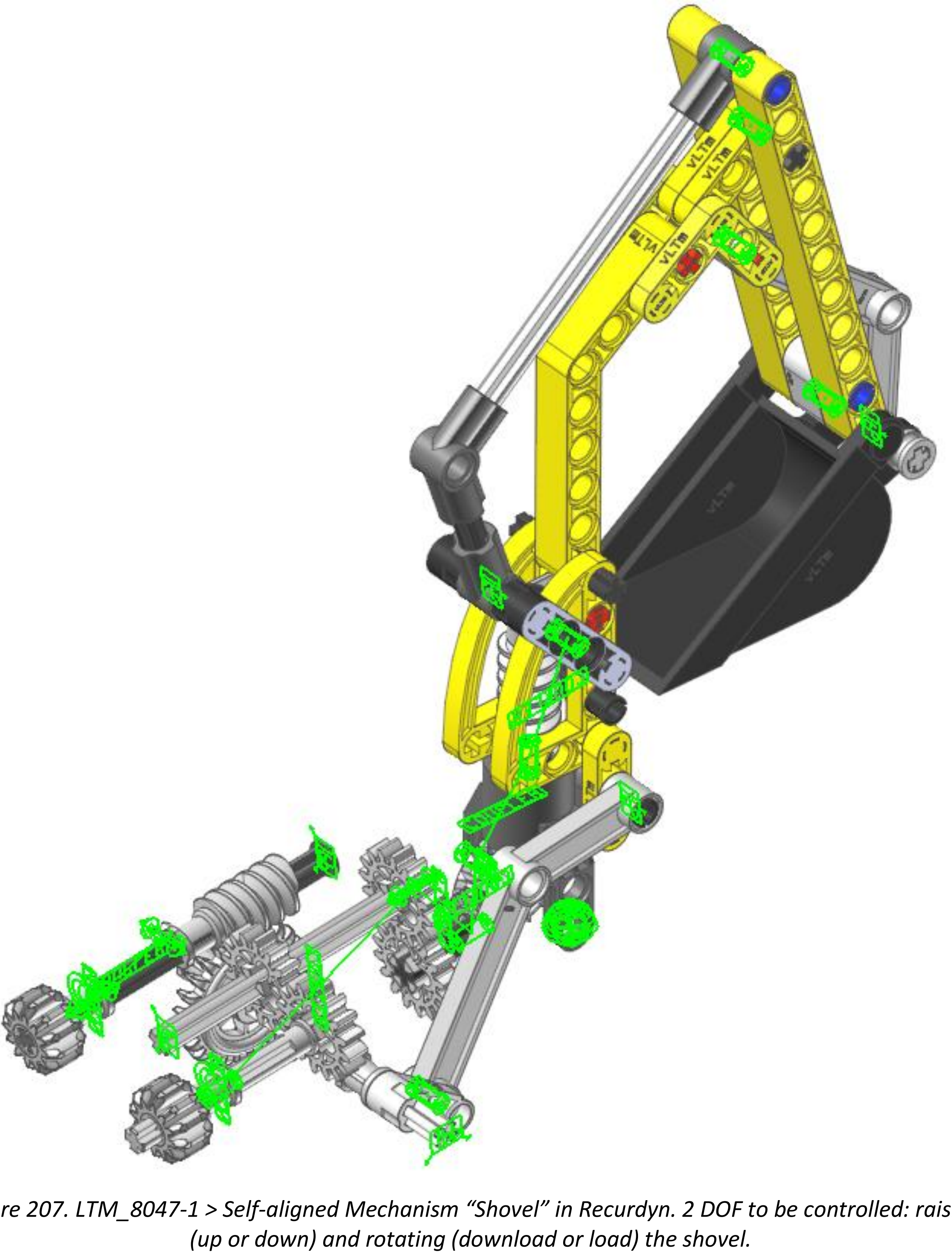


Figure 207. LTM_8047-1 > Self-aligned Mechanism "Shovel" in Recurdyn. 2 DOF to be controlled: raising (up or down) and rotating (download or load) the shovel.

Figure 208. LTM_8047-1 > A real life machine (5/11). > Shovel.

Figure 209. LTM_8047-1 > A real life machine (6/11). > Basket > Joints.

Figure 210. LTM_8047-1 > A real life machine (7/11). > The 4-bar used to control the inclination of basket with respect to chassis of the cylinders. It is a planar mechanism.

Physically, the two parallel horizontal bars of the figure are independent pieces; but a fixed (F) joint is created in the CAE program to properly calculate mobilities. However, red piece is just one piece.

Next figures, 8-11 /11, are a detail of 4/5 joints which can be seen in figure 7/11.

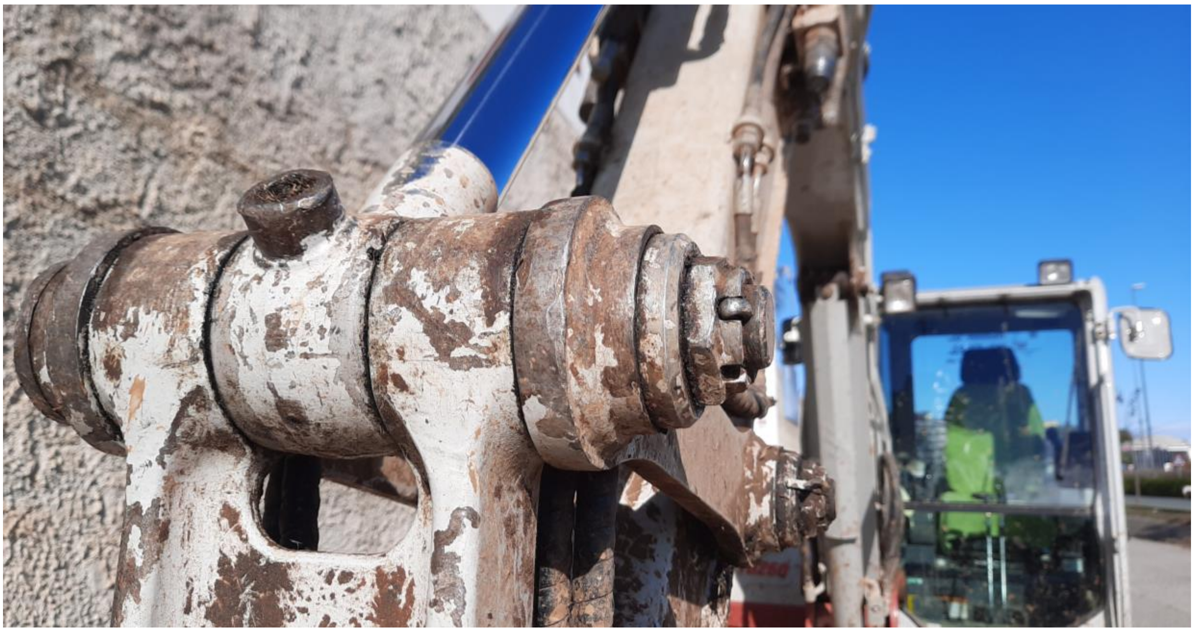


Figure 211. LTM_8047-1 > A real life machine (8/11). > A detail of one joint (1/4): sides.

Figure 212. LTM_8047-1 > A real life machine (9/11). > A detail of one joint (2/4): center.

Figure 213. LTM_8047-1 > A real life machine (10/11). > A detail of one joint (3/4).

Figure 214. LTM_8047-1 > A real life machine (11/11). > A detail of one joint (4/4).

1.4.3. SIMULATION: VLTM_8047-1

vLTm_8047-1_2018-wb-ch-31-All-iso1.av

vLTm_8047-1_2018-wb-ch-31-All-iso2.av

1.5. MODEL #3: LTM_ 8294-1

It is a **crawler excavator**.

1.5.1. ASSEMBLY: LTM_ 8294-1

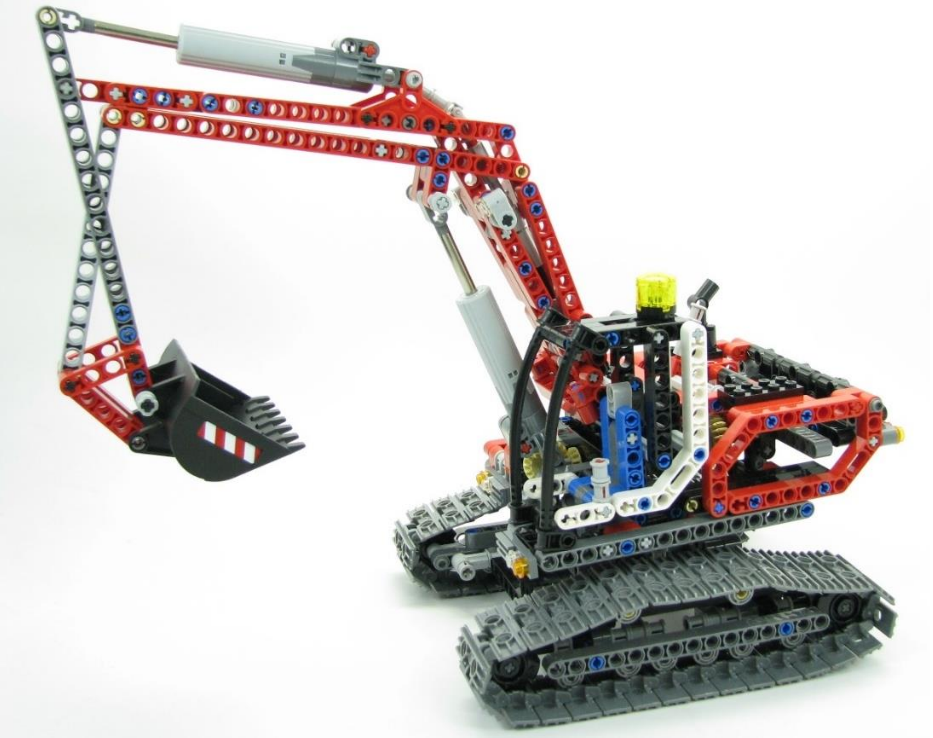


Figure 215. LTm_8294-1 in real life. Source: this is the source.

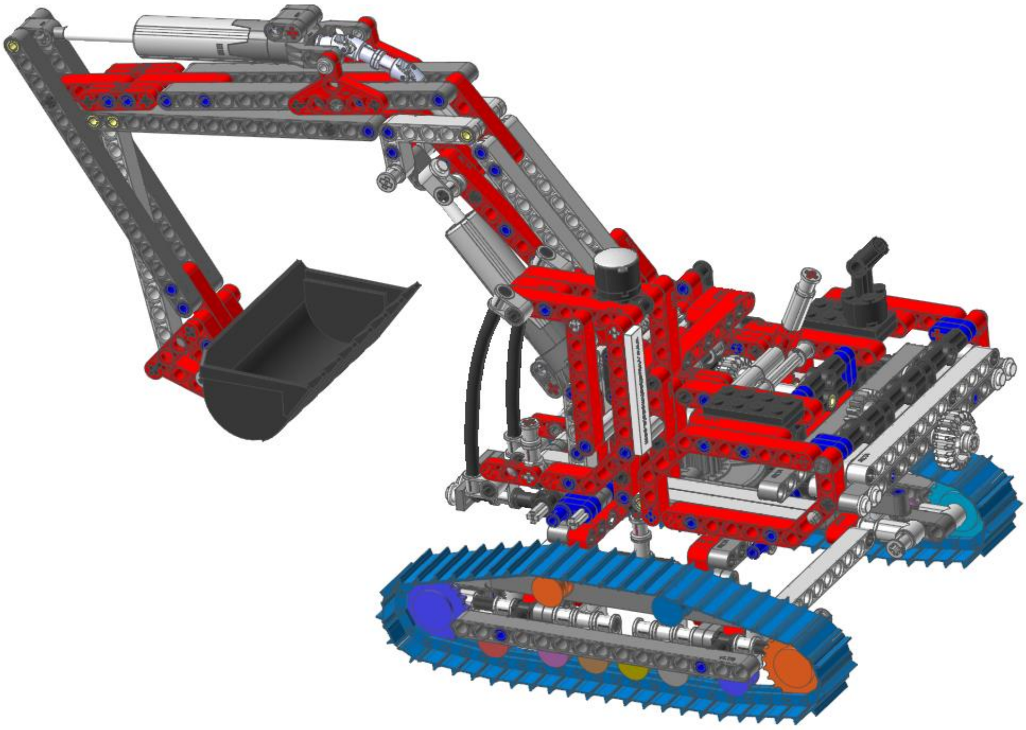


Figure 216. LTm_8294-1 in Recurdyn after using the "Track_LM" toolkit.

Figure 217. LTM_8294-1 > A real life machine (1/5). Source: this is the source (bossmachinery.nl).

1.5.2. MECHANISMS: VLTM_ 8294-1

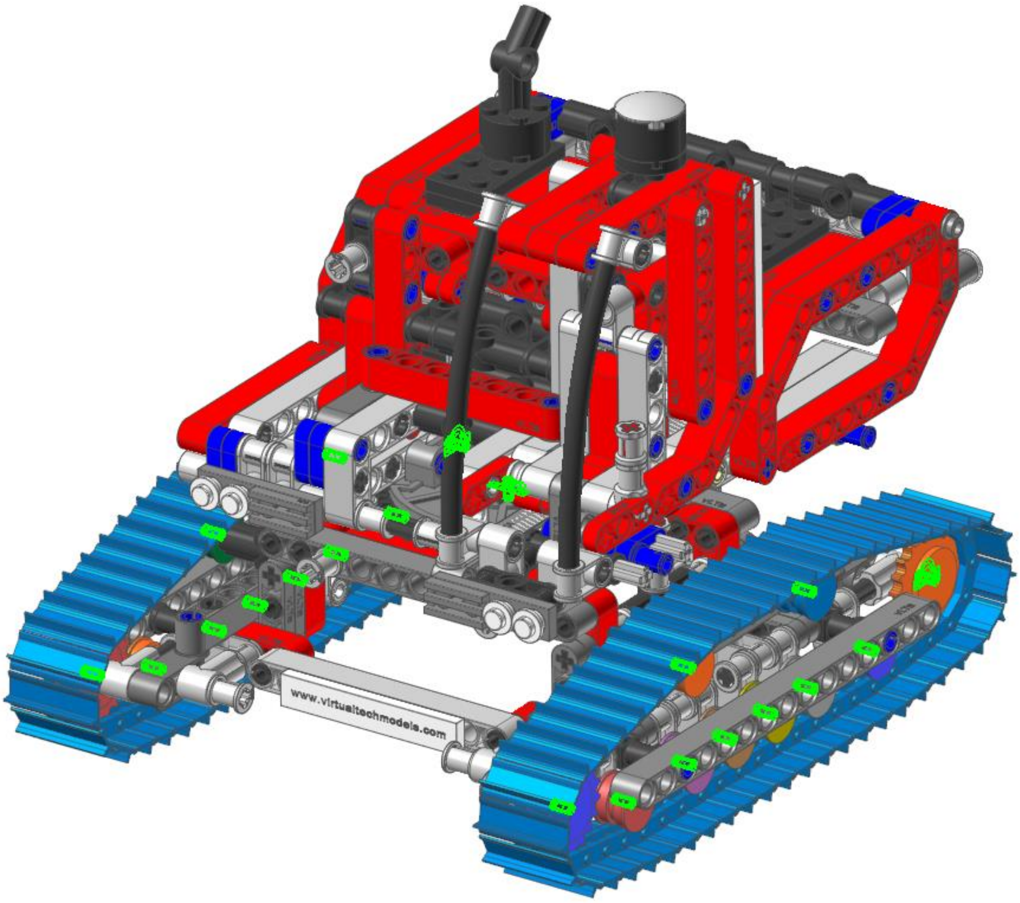


Figure 218. LTM_8294-1 > Self-aligned Mechanism "Driving chains and chassis' rotation" in Recurdyn. 3 DOF to be controlled: 2 driving chains and rotation of the top chassis with respect to the bottom chassis.

Figure 219. LTM_8294-1 > A real life machine (2/5). Source: this is the source (bossmachinery.nl).

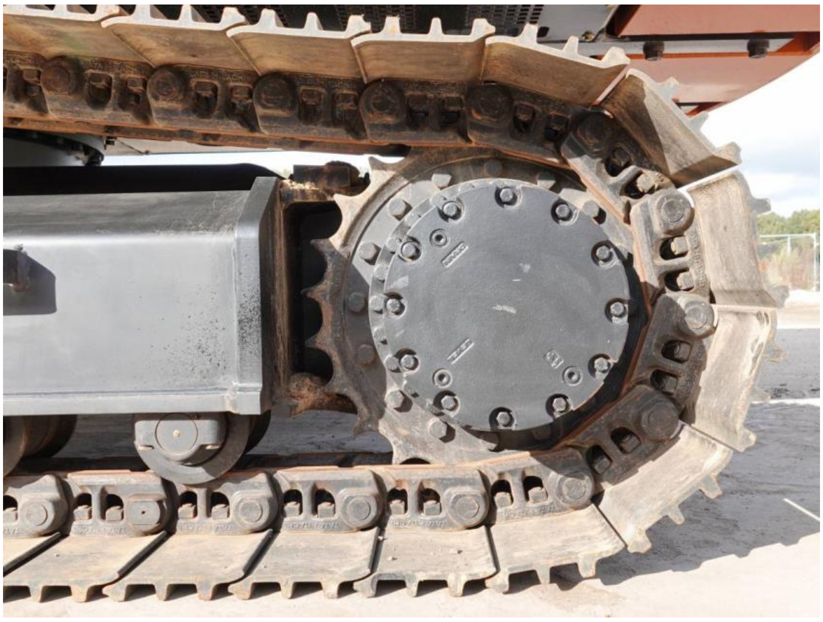


Figure 220. LTM_8294-1 > A real life machine (3/5). Source: this is the source (bossmachinery.nl).

Figure 221. LTM_8294-1 > A real life machine (4/5). Source: this is the source (bossmachinery.nl).

Figure 222. LTM_8294-1 > A real life machine (5/5). Source: this is the source (bossmachinery.nl).

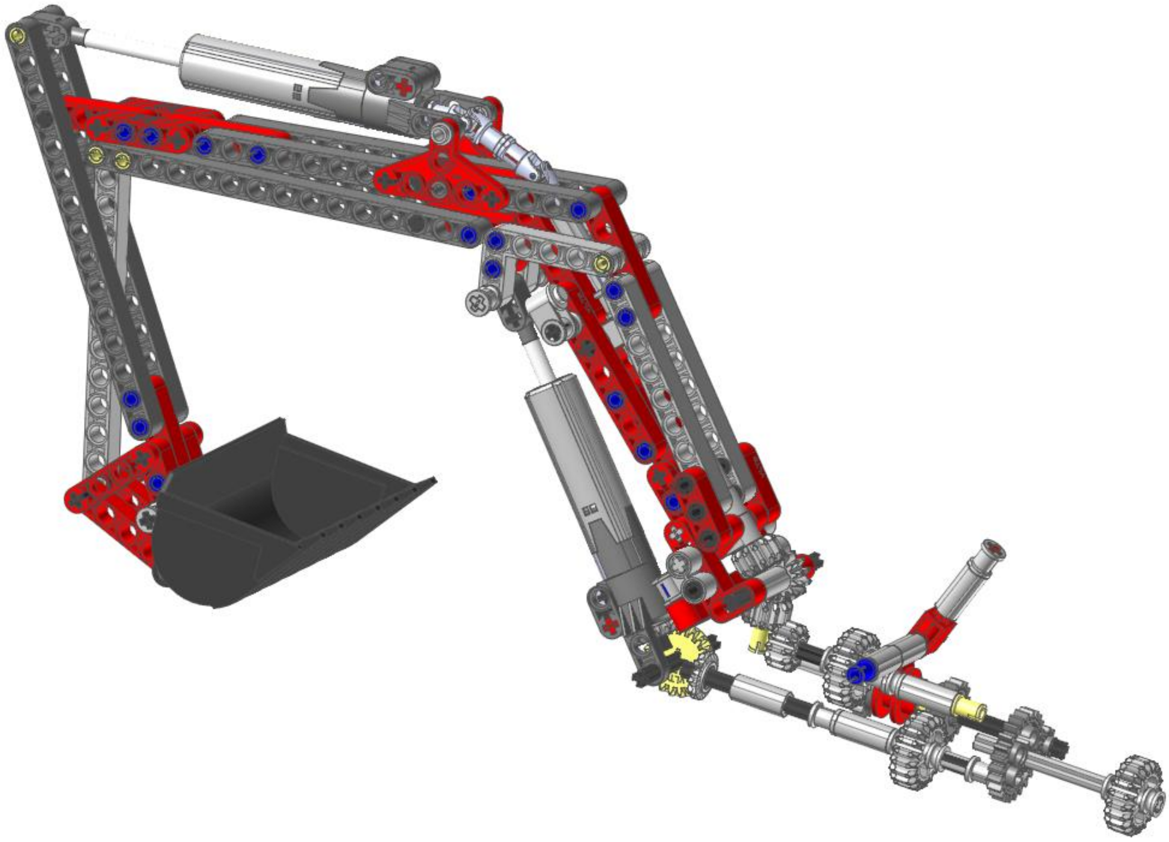


Figure 223. LTM_8047-1 > Mechanism “Shovel” in Recurdyn. 2 DOF to be controlled: raising (up or down) and rotating (download or load) the shovel.

1.5.3. SIMULATION: VLTM_ 8294-1

vLTm_8294-1-21_2019-wb.avi

1.6. MODEL #4: LTM_42124-1

It is a **buggy**.

1.6.1. ASSEMBLY: LTM_42124-1

Figure 224. LTm_42124-1 in real life. Source: this is the source (www.lego.com/es-es/).

Figure 225. LTm_42124-1 in SolidWorks® 2007.

Figure 226. LTM_42124-1 > A real life machine: Ariel Nomad. Source: top, : middle up, middle down, bottom (arielmotor.co.uk).

1.6.2. MECHANISMS: VLTM_42124-1

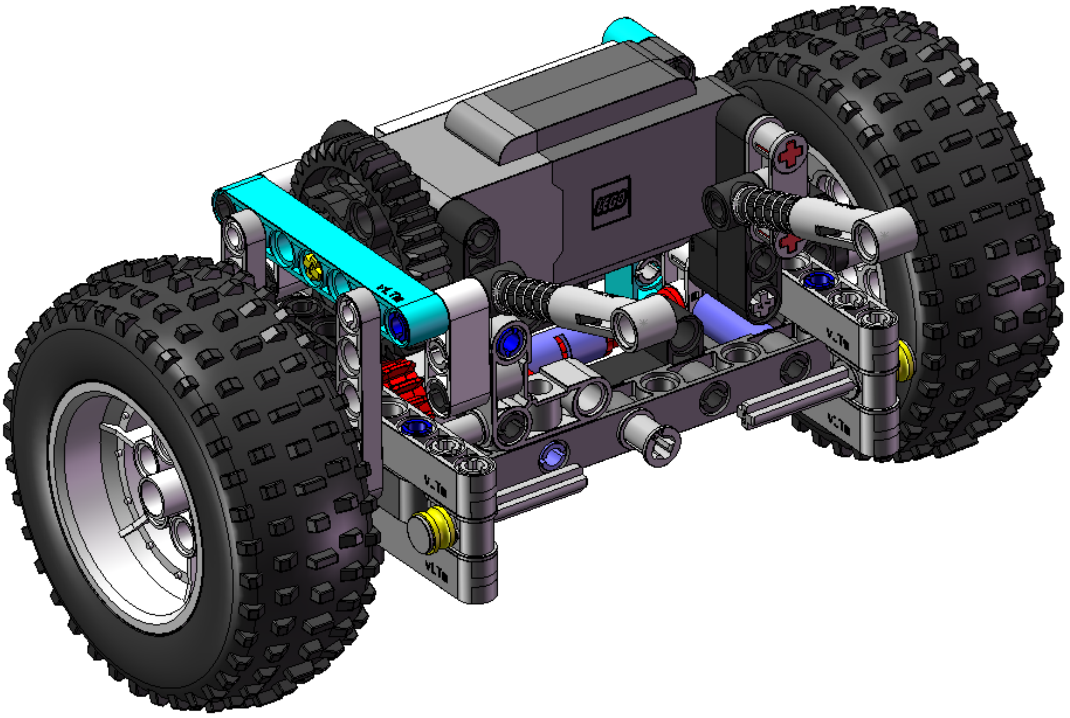


Figure 227. LTm_42124-1 > Mechanism "Power" (P) in SolidWorks® 2007. 3 DOF: 2 driving wheels and 1 independent movement from shock-absorbers (both shock-absorber have the same motion). Or 2 DOF: axis from engine and the independent movement.

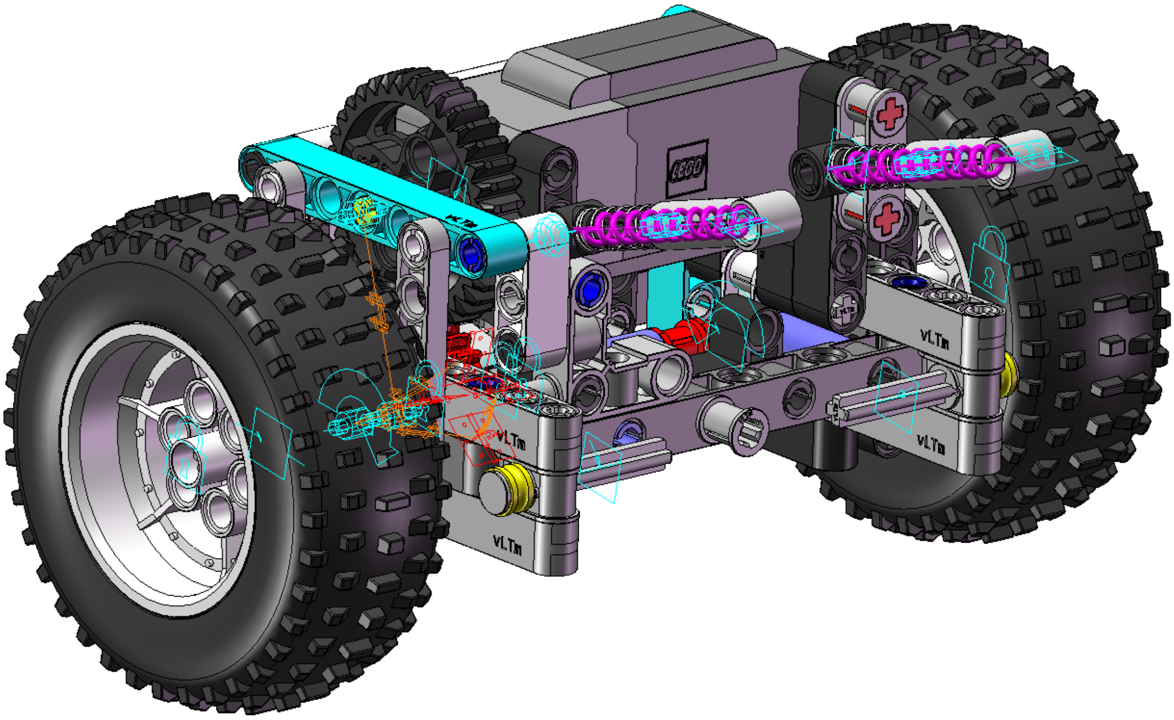
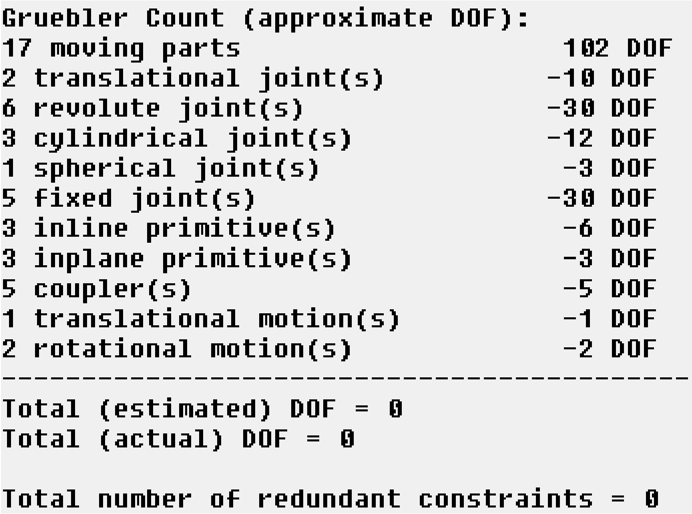


Figure 228. LTm_42124-1 > Self-aligned "P" in COSMOSMotion 2007. Damping system is recreated by using "springs" and "damping" (forces).



```
Gruebler Count (approximate DOF):
17 moving parts                     102 DOF
2 translational joint(s)            -10 DOF
6 revolute joint(s)                 -30 DOF
3 cylindrical joint(s)              -12 DOF
1 spherical joint(s)                 -3 DOF
5 fixed joint(s)                    -30 DOF
3 inline primitive(s)                -6 DOF
3 inplane primitive(s)               -3 DOF
5 coupler(s)                         -5 DOF
1 translational motion(s)            -1 DOF
2 rotational motion(s)               -2 DOF
-------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 229. LTm_42124-1 > Self-aligned "P" > COSMOSMotion 2007 Simulation Panel.

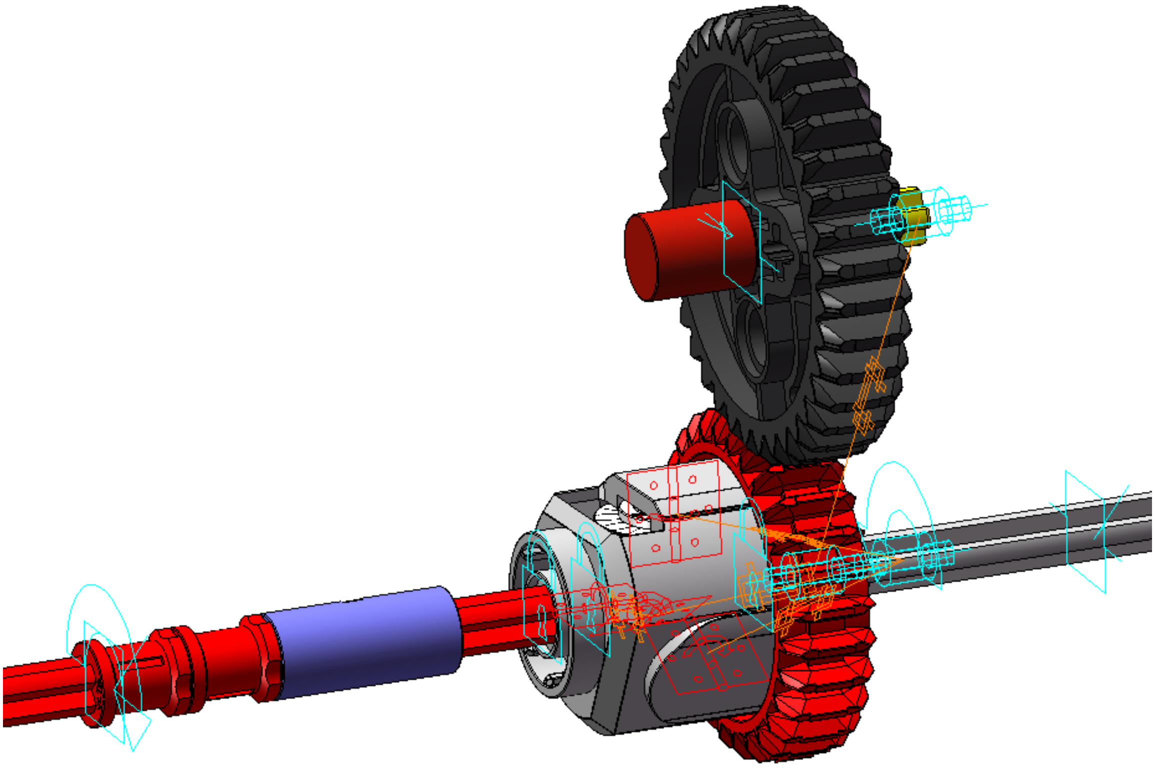


Figure 230. LTm_42124-1 > Self-aligned "P" in COSMOSMotion 2007 > A detail of how power fluxes from the engine to the wheels.

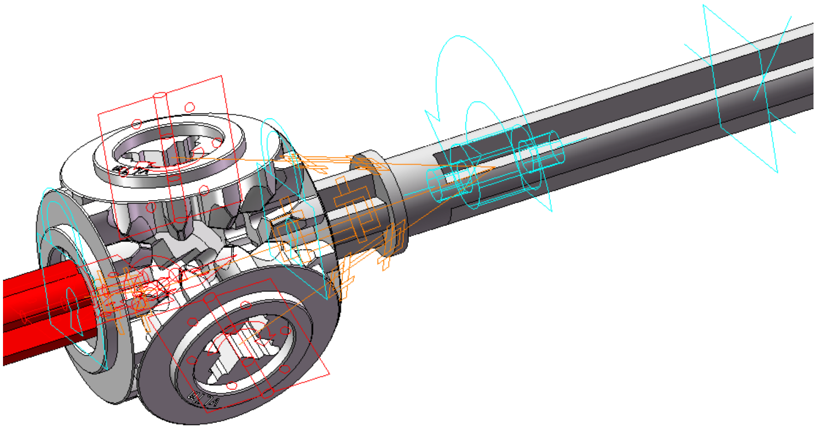


Figure 231. LTm_42124-1 > Self-aligned "P" in COSMOSMotion 2007> The inside of the differential.

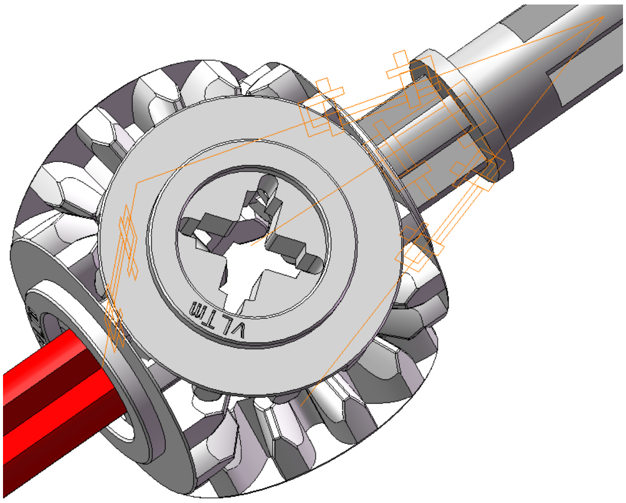


Figure 232. LTm_42124-1 > Self-aligned "P" in COSMOSMotion 2007 > Four couplers are needed to define how gears move within the differential.

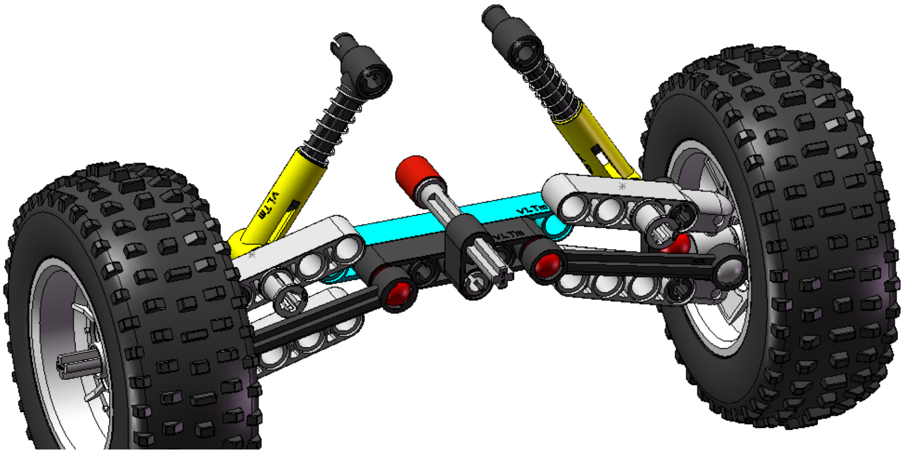
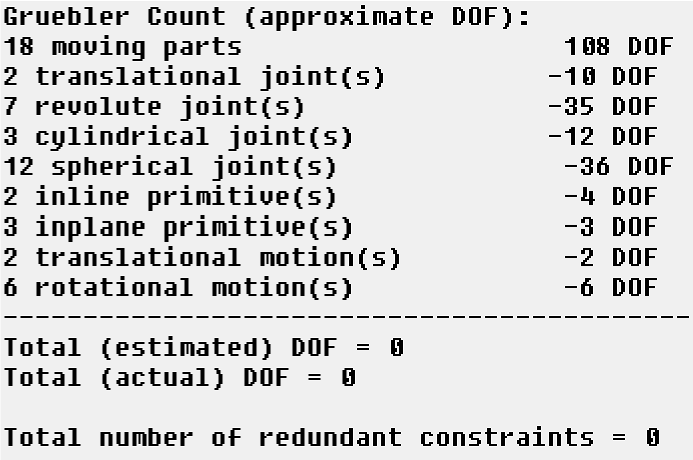


Figure 233. LTm_42124-1 > Mechanism "Direction" (D) in SolidWorks® 2007

Figure 234. LTm_42124-1 > Self-aligned "D" in COSMOSMotion 2007. 5 DOF: 2 independent shock absorbers and 2 wheels (4 passive DOF) and left-right control (1 DOF to be controlled).



```
Gruebler Count (approximate DOF):
18 moving parts                    108 DOF
2 translational joint(s)          -10 DOF
7 revolute joint(s)               -35 DOF
3 cylindrical joint(s)            -12 DOF
12 spherical joint(s)              -36 DOF
2 inline primitive(s)              -4 DOF
3 inplane primitive(s)             -3 DOF
2 translational motion(s)          -2 DOF
6 rotational motion(s)             -6 DOF
-------------------------------------------
Total (estimated) DOF = 0
Total (actual) DOF = 0

Total number of redundant constraints = 0
```

Figure 235. LTm_42124-1 > Self-aligned "D" > COSMOSMotion 2007 Simulation Panel.

Figure 236. . LTM_42124-1 > A real life machine: Ariel Nomad. Source: this is the source (arielmotor.co.uk).

1.6.3. SIMULATION: VLTM_42124-1

vLTm_42124-1_2021
-wB-motion-base-sp

vLTm_42124-1_2021
-wB-motion-base-sp