

KISSsoft AG - ☎ +41 55 254 20 50
 Uetzikon 4 - ☎ +41 55 254 20 51
 8634 Hombrechtikon - ✉ info@KISSsoft.AG
 Switzerland - www.KISSsoft.AG

Bevel differential

1 Description

1.1 Task

Bevel gear differentials can be modelled in KISSsys, but the procedure is bit difficult and may need some time to be able to model it correctly. This paper is written to explain the modelling procedure in principle. Because of the large variation of the differential designs all special things cannot be described here. In case if you need some more instructions, please contact KISSsoft AG and ask for KISSsys support.

You may also review the corresponding KISSsys model “Bevel-gear-differential-simple.ks”

1.2 Schematic

The following picture explains the basic structure of the model used in this report. Input is given for the pinion shaft “s1” and the shaft “s2” is the differential housing and is supported to the gearbox housing. Inside the differential housing are the differential gears so that “sd” is connected to the differential housing and the gears “sd2” are on the driving shafts “s3” and “s4”. These shafts are again supported to the gearbox housing.

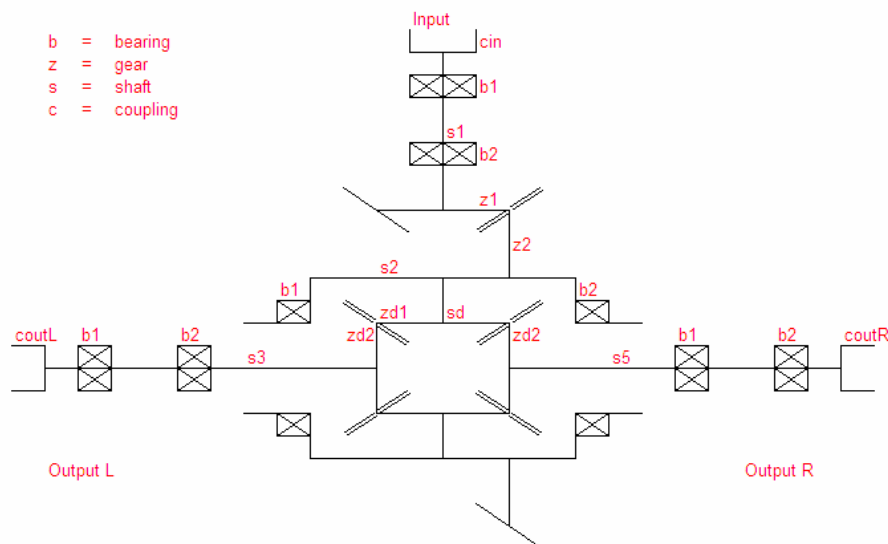


Figure 1.2-1 Schematic of the differential model

2 Modelling

2.1 Create a model

Create a model as any other model. Add all machine elements and connections between the gears. For the bevel gear connection use connection type “kSysGearPairConstraint” from the “templates” and for the bevel gear differential connections use “kSysPlanetaryBevelGearConstraint”.

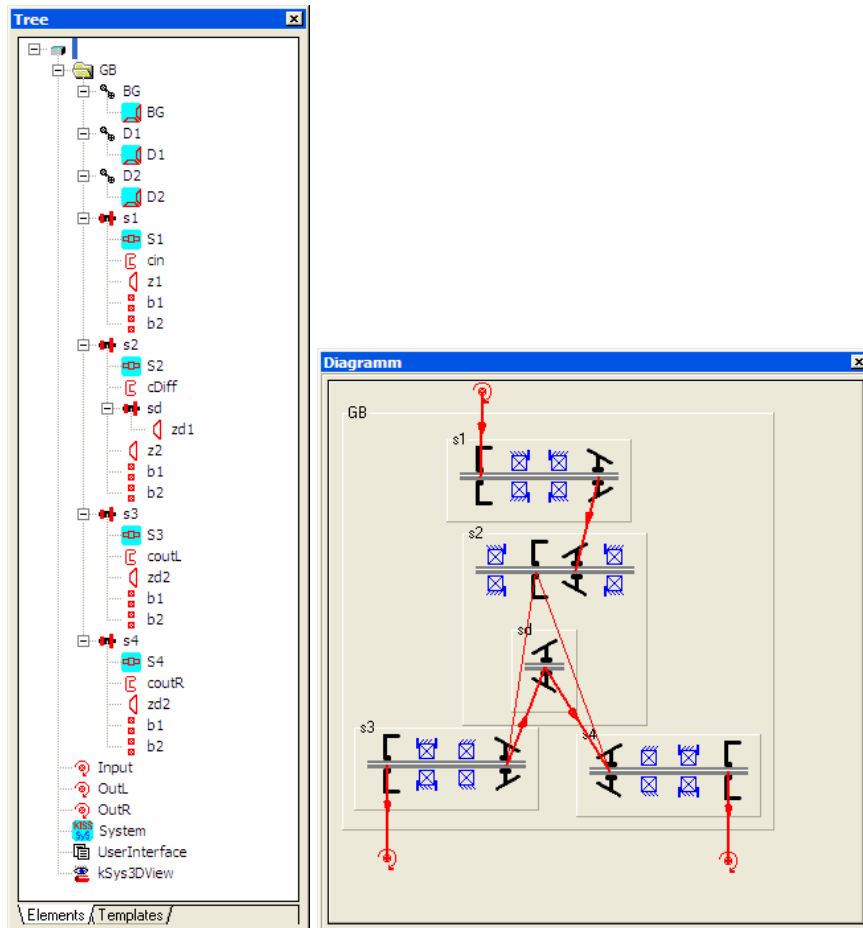


Figure 2.1-1 Model tree and schematic

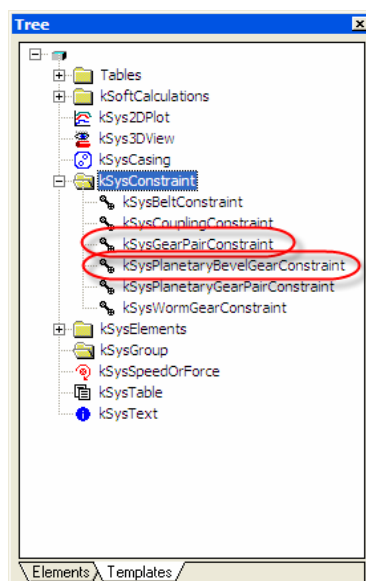


Figure 2.1-2 Gear connections for the differential gears

2.2 Model specialties

The following chapters explain special arrangements needed for the differential modelling.

2.2.1 “Carrier” definition

You need one extra component to the model to tell the program how differential gears are connected to the housing and which shaft is the differential housing and also how many differential planets are in the model. To do this, use “kSysPlanetCarrierCoupling” from the templates. This component defines the configuration of the differential.

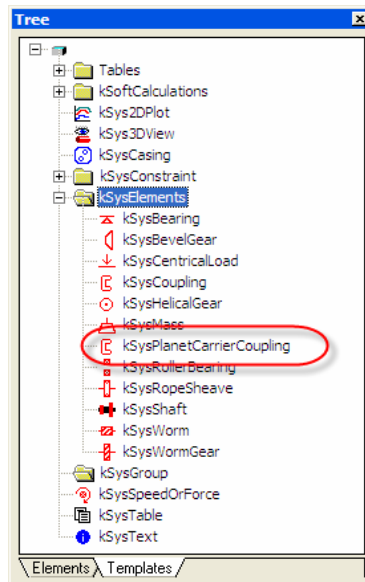


Figure 2.2-1 Differential carrier component

After you have placed this component in to your model on your differential housing “s2” you can define number of planets in differential by selecting “properties” for that component and changing the value for the variable “NoPlanets”. If you change the number of the planets you need to “updateShaftElements” to consider new number of the planets for the calculations.

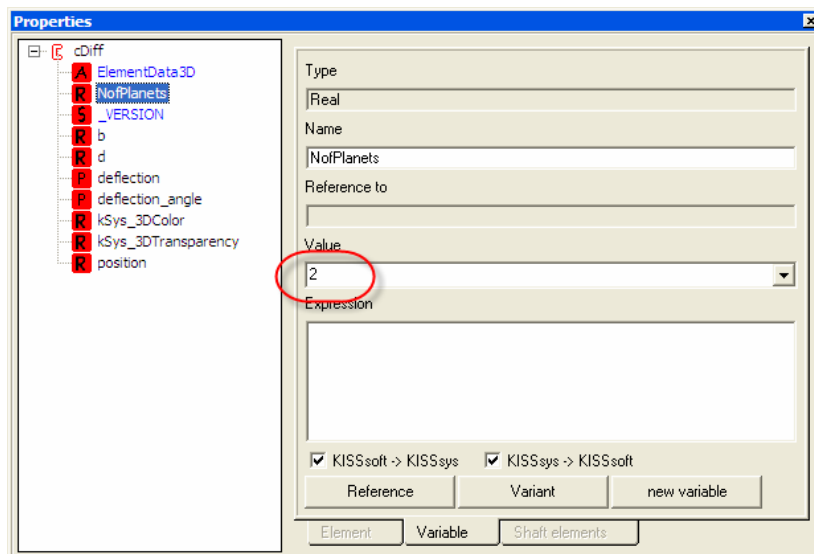


Figure 2.2-2 Changing the number of differential planets

Note! In case if you are using old templates where this component is not present, you can use “kSysCoupling” component instead and add needful information in it. Create new variable

“NofPlanets” in the component and it has the same functionality. Remember to add flags for both directions.

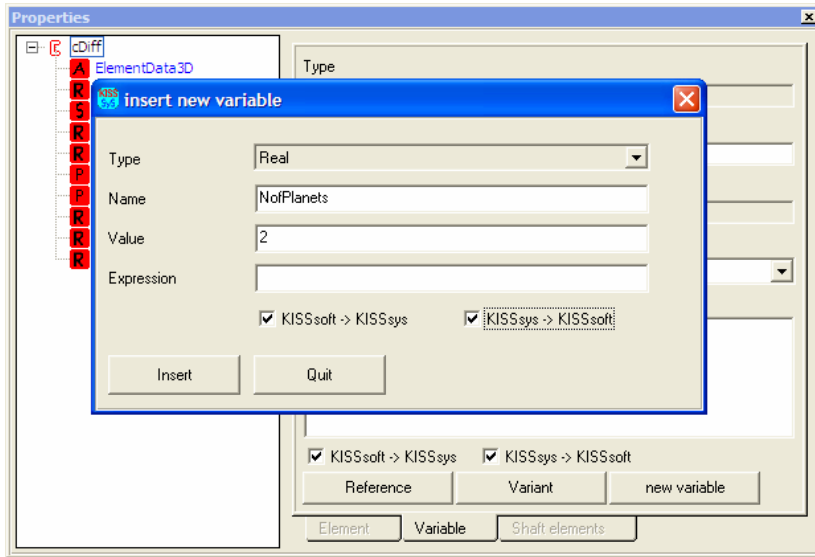


Figure 2.2-3 If using "kSysCoupling" add a new Real variable called "NofPlanets"

2.2.2 Differential gear connections

Use “kSysPlanetaryBevelGearConstraint” to define connections between differential bevel gears. You need to define first configuration from the list “gear/planet” or “planet/gear”. First gear in the definition is the “gear 1” and second is the “gear 2”. You need to make the selection so that the first gear has less number of teeth. From the configuration “gear” means the output gear and “planet” means the gear connected to the differential housing. You need to also define which coupling is the “planet carrier”.

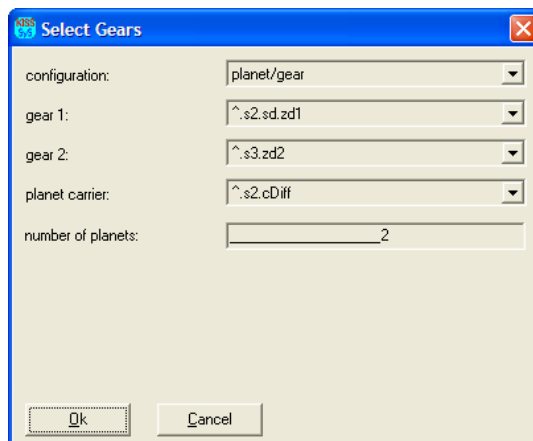


Figure 2.2-4 Create a connection between differential gears

2.2.3 Kinematics Iteration

The iterations for the speed and torque is not activated as per default. To activate iterations, right click on “System” and choose “Properties”. Check the variable “kSysKinematicMode”. Select “iteration for speed and torque” from the list. Model may work correctly even without this iteration method, but when model gets more complicated this iteration is needed.

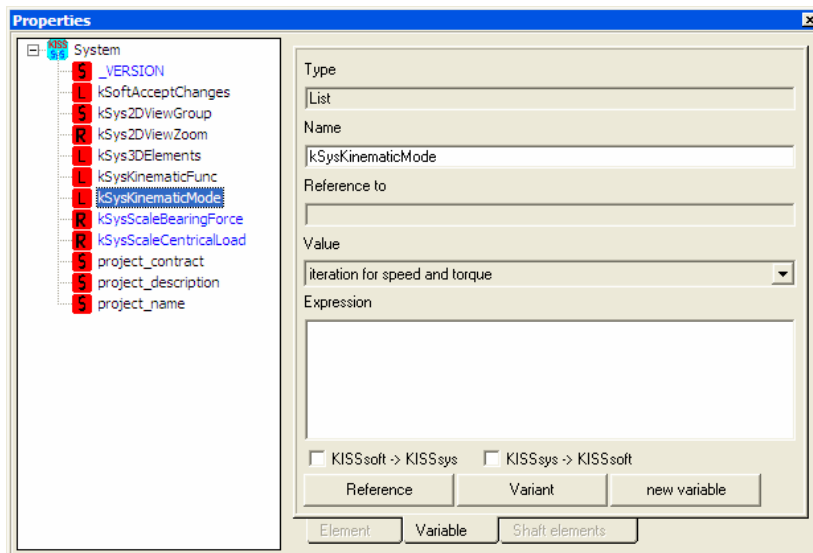


Figure 2.2-5 Select iteration method for the kinematic calculation

3 Gear positioning

3.1 Positioning according to bevel gears

To be able to make “kinematic” calculation user needs to define correct directions and positions of the output shafts in the space, because otherwise it is not possible to define output speeds and torques to the correct directions and error in “kinematic” calculation will appear. Differential gears needs to be therefore defined according to each other, so that correct speeds can be defined for all the shafts.

It is recommended to define first the position of the differential housing “s2”. Then define other of the outputs “s3” or “s4” to be parallel to the “s2”. Then define differential planet “sd” position according to the differential gear meshing with gear in the defined shaft and finally define the direction and position of the other shaft meshing with the differential planet gear.

Use dialog for all shaft to make the positioning!

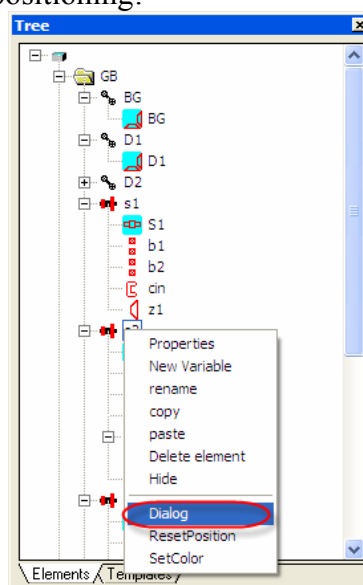


Figure 3.1-1 Choose "dialog" function from the tree to define position for the shafts

For the bevel gear connection choose “According Bevel Gears” from the list

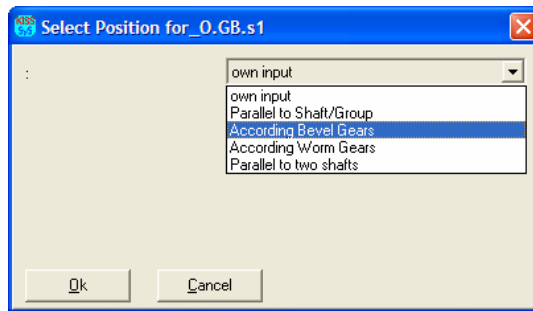


Figure 3.1-2 Choose correct positioning method from the list

- 1) Define the position of the “s2” “According Bevel Gears”. Select mating gear and define also correct contact angle.

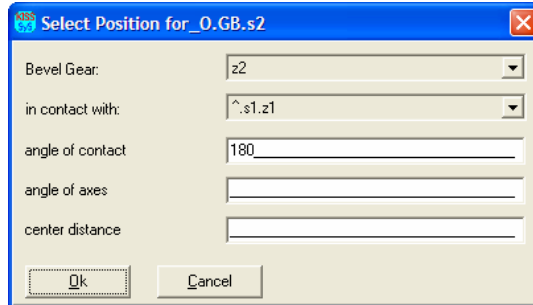


Figure 3.1-3 Positioning of the shaft "s2" according to the bevel gear mesh “z2” and “z1”

- 2) Define “s4” position to be parallel to “s2”

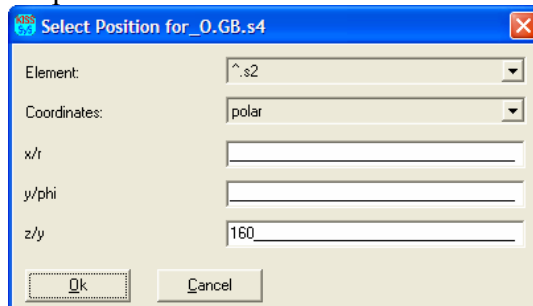


Figure 3.1-4 Right output "s4" defined parallel to differential housing "s2"

- 3) Define differential planet “sd” according to mating bevel gear in shaft “s4”

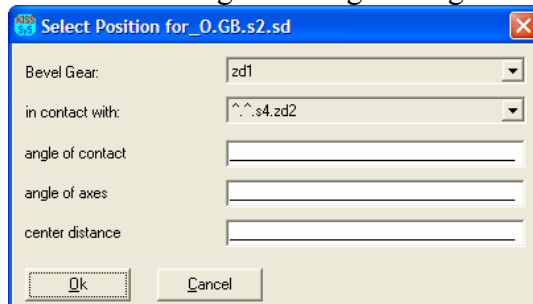


Figure 3.1-5 Planet shaft defined according to bevel gear mesh

- 4) Finally define direction and position of the left output “s3” according to bevel gear mesh with planet gear.

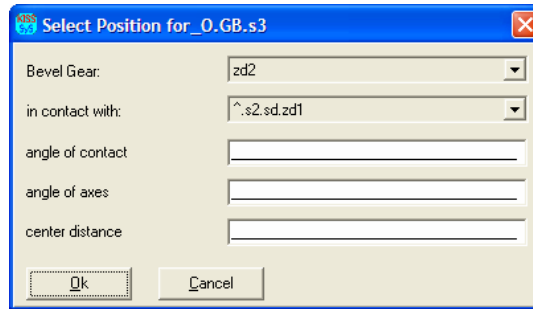


Figure 3.1-6 Left output "s3" defined according to bevel gear mesh

4 Boundary conditions

4.1 Input

Because there are six conditions to define for inputs and outputs (speed + torque/power), we need to define three of them to be able to run the calculation. Speed and torque definition of a bevel gear differential has to be made correctly and usually input is totally defined and output is then calculated.

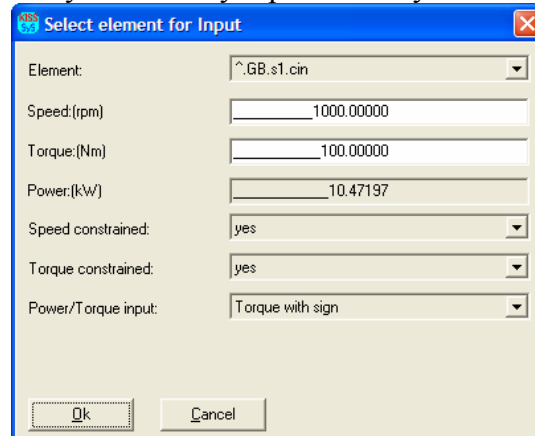


Figure 4.1-1 Input fully constrained

4.2 Outputs

Now the state of the two outputs is undefined. One useful option would be the definition of the speed of one differential output (example: s3) related to the casing speed with a factor k:

$$k = \frac{n_{s3}}{n_{s2}}$$

This makes it possible to define a formula for the calculation of the left output speed in our model as follows:

$$n_{out} = n_{s3} = kn_{s2}$$

Definition of the output in the example model has to be made as follows:

1. Definition of global variable "k" type "real"
2. Setting the "kSysSpeedOrForce" element of the left output to "Speed constrained: yes"
3. Entering the expression for the correct speed calculation into the variable "speed" of the "kSysSpeedOrForce" element of the left output.

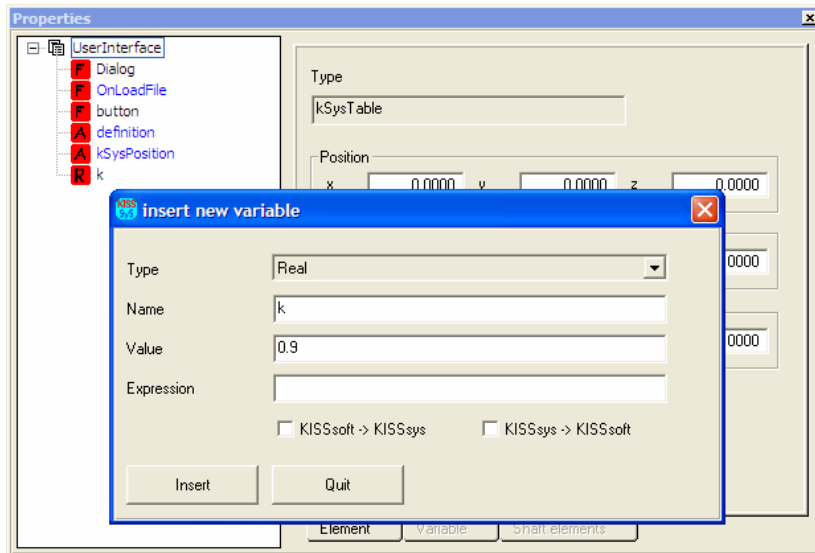


Figure 4.2-1 Define new variable "k"

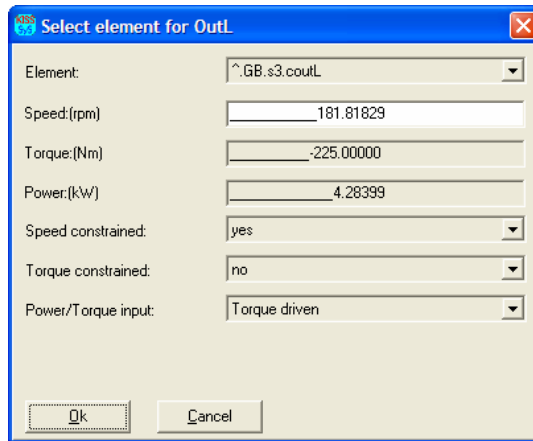


Figure 4.2-2 Constrain the speed value

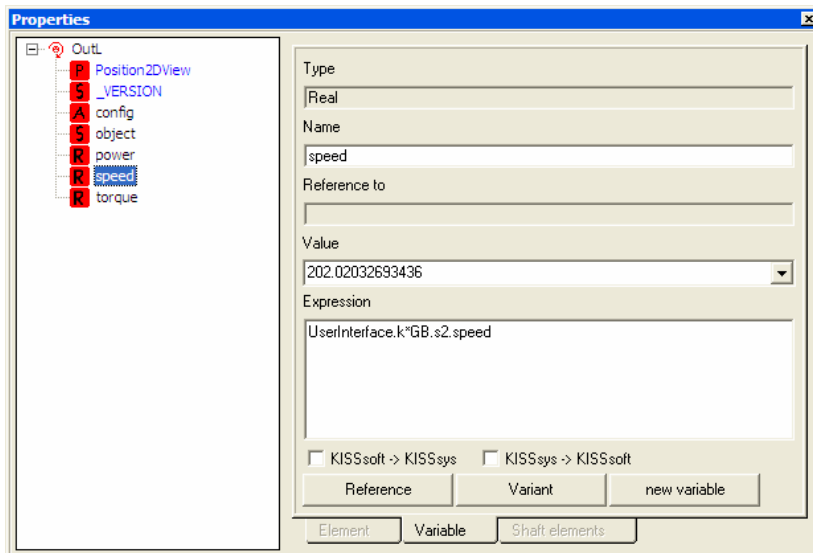


Figure 4.2-3 Make expression for the "OutL" speed

Attention: Expressions of the element variables are deleted every time you open the dialog of a "kSysSpeedOrForce".

5 Calculations

5.1 Differential bevel gears

Recommended calculation method for the bevel gears in differential arrangement is “Static calculation”. This means that even if there is some speed difference inside the differential calculation is based on the static calculation with maximum possible torque. This torque needs to be defined manually for the calculations and is not taken from the values in KISSsys.

5.2 “Simplification”

In case if you are not interested any calculations inside differential, it is recommended not to model it at all, but to use even more simplified method considering differentials as black box and to divide only torque between two outputs having equal speed.