

Changements apportés sur le modèle AxleSplitHEVv6 durant le BEI 2017-2018

Durant le BEI nous avons modifié en partie le modèle mis en place durant les années précédentes, les modifications sont faites jusqu'au niveau des blocs qui seront auto codés, à l'intérieur de ces blocs aucun changement n'a été fait sauf cas exceptionnel. Tous les changements apportés au modèle sont détaillés dans ce document. Pour connaître les blocs qui ont été utilisés pour l'auto codage, veuillez-vous référer au document concernant l'architecture des processeurs.

Nous ne détaillerons donc que les blocs visant à être auto codés ainsi que les blocs dans lesquels ils sont contenus. Les flèches indiquent la hiérarchie dans les blocs.

Bloc Platform : Toutes les anciennes entrées et sorties ont été changées pour mettre une logique de bus (bus selector/bus creator), de même à l'intérieur du bloc lui-même.

Bloc Electrical System : Toutes les anciennes entrées et sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc Lithium – Ion Battery Management System** : La partie suivante a été intégrée au bloc dans l'optique de faire l'auto codage. L'entrée *SYSE_LIION_CURRENT* a été remplacée par les deux entrées *RA_ME_CURRENT* et *SYSE_AUX_CURRENT* le calcul présent ci-dessous ce faisant à l'intérieur du bloc.

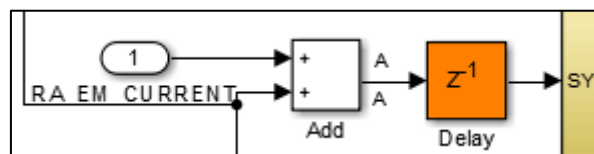


Figure 1 - Partie de calcul intégrée dans le bloc

On a ensuite remplacé les entrées et les sorties de ce bloc pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc Auxiliary network 12V** : Le bloc étant mono-entrée, mono-sortie, rien n'a été modifié sur celui-ci.

↳ **Bloc Electrical System supervision** : Toutes les anciennes entrées et sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

Bloc Command : Toutes les anciennes entrées et sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

Bloc Reference (NEDC) : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

Bloc Driver1 : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

Bloc Front Axle : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc Clutch** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même. A l'intérieur du bloc nous avons également fait une modification. En effet un bloc multiplieur possédait un port qui n'était pas connecté, l'autre unité de travail nous a informés qu'au vu des équations, la connexion devait être faite. Nous l'avons donc fait de la manière suivante (fil surligné) :

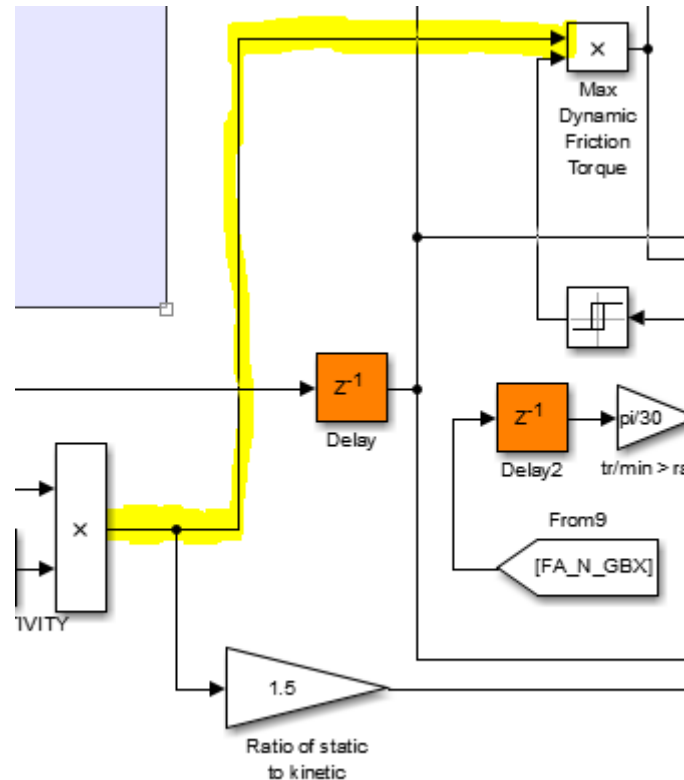


Figure 2 - Reconnexion du bloc multiplieur

↳ **Bloc Gearbox + Differential** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc ICE** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même. De plus nous avons remplacé l'entrée *ground* par une entrée constante nulle, ce qui est plus convenable pour l'auto codage.

↳ **Bloc ICE control** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc ICE** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc FA_TQ_to_F_BRAKE** : Le bloc étant mono-entrée, mono-sortie, rien n'a été modifié sur celui-ci.

Bloc Rear Axle1 : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc Differential** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc *RA_TQ_to_F_BRAKE*** : Le bloc étant mono-entrée, mono-sortie, rien n'a été modifié sur celui-ci.

↳ **Bloc *Electrical Machine*** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc *Engine representation*** : Toutes les anciennes sorties ont été changées pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

↳ **Bloc *Power efficiency*** : La partie suivante a été intégrée au bloc dans l'optique de faire l'auto codage. Une entrée *SYSE_VOLTAGE_HV* a été rajoutée au bloc pour pouvoir faire le calcul à l'intérieur, la sortie *RA_EM_POWER* est donc remplacée par la sortie *RA_EM_CURRENT*.

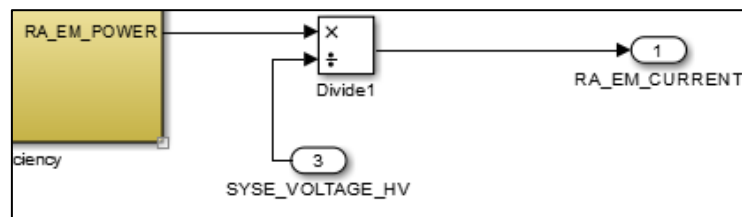


Figure 3 - Partie intégrée dans le bloc

On a ensuite remplacé les entrées et les sorties de ce bloc pour mettre une logique de bus (bus selector / bus creator), de même à l'intérieur du bloc lui-même.

Bloc *Environment* : Bloc mono-sortie, rien n'a été modifié.

Modification sur d'autres blocs :

Une partie du bloc *Command* : *Command torque towheel* → *torque to clutch conversion FA* à également été modifiée. En effet, lorsque nous lançons l'ancien modèle nous avons un warning qui pourrait potentiellement mener à un crash dans l'auto codage. Le warning était le suivant :

Warning: Division by zero in 'AxleSplitHEVv6Old/HEV model/Command/Command - Torque towheel->torque to clutch conversion FA/Divide2'

Cela signifie que parfois, une division par zéro est effectuée. En effet, regardons où nous indique le warning :

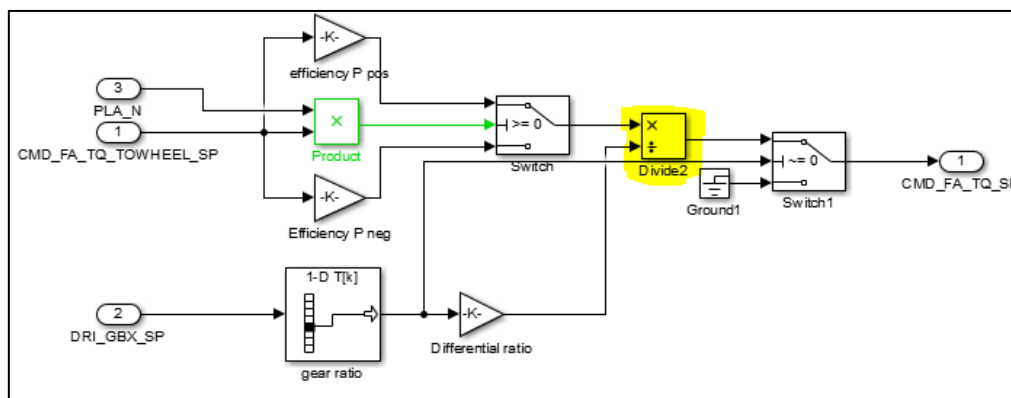


Figure 4 - Endroit d'où est issu le problème

Le problème intervient au niveau du bloc *Divide2*, en effet des divisions par zéro ont lieu comme on peut le voir en observant la sortie du bloc *gear ratio* :

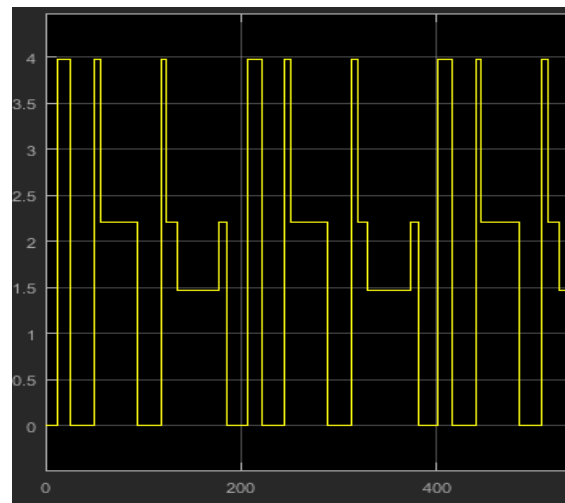


Figure 5 - Sortie du bloc *gear ratio*

On voit bien que la sortie passe souvent par 0 et que donc des divisions par zéro se font. Heureusement ce calcul n'est pas pris en compte grâce au switch avec le bloc de division qui permet d'envoyer 0 si on a 0 en sortie du *gear ratio*. Il n'empêche que la division par zéro est quand même faite. On ne peut tolérer ça lors de l'auto codage nous avons donc trouvé un moyen de contourner ce problème grâce à la constante matlab *eps* qui correspond à une très petite valeur. Voici la solution mise en place.

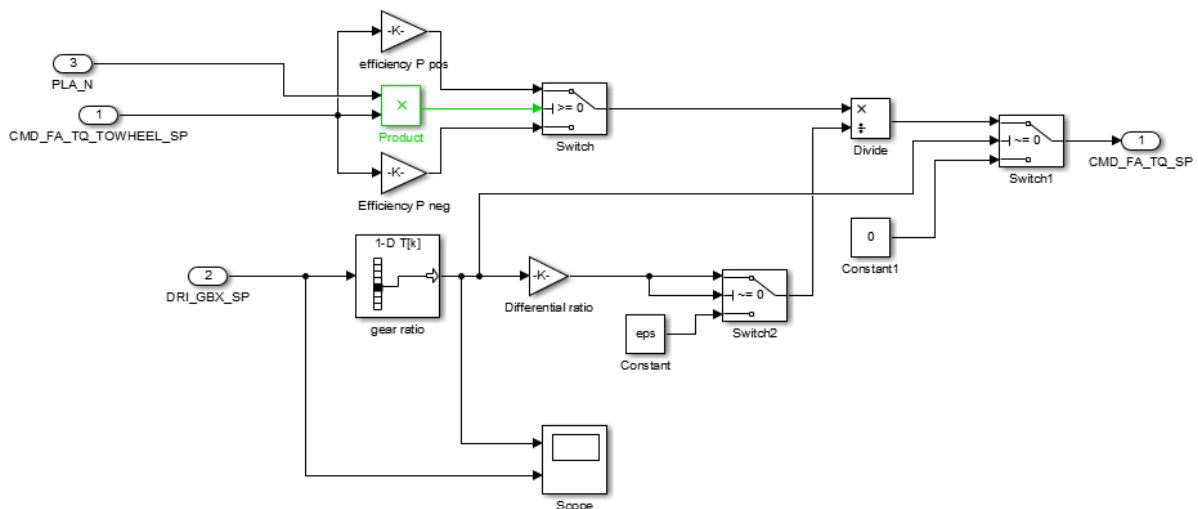


Figure 6 - Solution proposée pour le problème de division par 0

En sortie du *differential ratio* on met donc un switch qui prend la valeur de sortie si celle-ci est non nulle et *eps* si la sortie est nulle. On envoie donc ceci dans le bloc de division et ensuite on a un deuxième switch qui permet de ne pas prendre en compte la division si la sortie du *gear ratio* est nulle.