Explanation on Simulink architecture for Nova

# 1st level: Global architecture

Dit is de algemene structuur van de simulink file. Deze structuur bestaat uit 5 grote bouwblokken die we verder in detail gaan bespreken: setup, inputs, startup algorithm, processor en outputs.

# 1st level: Setup

In deze sectie worden messagebytes gemaakt in een matlab script. Deze worden dan doorgestuurd naar een CAN bus om het systeem volledig te initialiseren (setup).

# 1st level: Inputs

In deze sectie worden de gegevens van de sensoren, de 4 drives, de energiemeter, het batterijpakket en de commando’s op het dashboard verzameld. Deze worden dan doorgestuurd en opgeslagen in een outport en verder gebruikt in verschillende subonderdelen. Deze komen natuurlijk ook ergens vandaan, wat verder uitgelegd staat.

## 2nd level: Sensors

Een belangrijk onderscheid moet worden gemaakt tussen de sensoren vooraan en achteraan in de wagen. Diegene vooraan zorgen voor het opmeten van de throttle value, de brake value, de steering angle, de suspension rechts en de suspension links (de travel van de suspension wordt dus opgemeten, indien de stijfheden/dempingsconstanten gekend zijn kunnen daaruit de veer- en dempingskrachten volgen). Al deze elementen worden doorgestuurd naar een grote bus die al deze sensorinformatie zal bijhouden onder de naam ‘Front\_sensor’.

De achterste sensoren zorgen voor het opmeten van de yaw rate en de roll rate door middel van een yaw sensor (Bosh M55…) en de 3-dimensionale versnelling van de wagen door middel van een inertieel meetsysteem. Ook zijn de waarden van de rechtse en linkse suspension hier opgemeten. Al deze elementen worden doorgestuurd naar een grote bus die al deze sensorinformatie zal bijhouden onder de naam ‘Rear\_sensor’.

Daarnaast zijn er nog 2 temperatuursensoren, vooraan en achteraan in de wagen.

## 2nd level: Pre processor

Bij het uitlezen van sensoren is het altijd belangrijk om te weten in welke eenheden de grootheden zijn uitgedrukt. Deze zijn niet altijd in overeenkomst met de grootheden die gebruikt worden in de controle algoritmen, waardoor een pre processing moet gebeuren om de grootheden om te vormen in grootheden die wel gebruikt kunnen worden in de algoritmen. Dit gebeurt hier voor de brakes (delen door 65536), de throttle en de steering angle (geeft de travel van de rack weer als resultaat uit de potentiometer).

Vooraleer deze gegevens verder gaan, moeten we nog controleren of we binnen bepaalde limietwaarden blijven. Dit gebeurt voor de throttle. Een minimum en maximumwaarde wordt meegegeven en we controleren of de waarden van de 2 sensoren daar binnen vallen. Indien dat het geval is (geen error als resultaat), nemen we het gemiddelde van de twee throttle waarden om verder te werken. Deze throttlewaarde wordt nog met 100 vermenigvuldigd. Het resultaat is een ‘Front\_proc’, die alle getransformeerde sensorwaarden bevat om verder in het algoritme te gebruiken.

Hetzelfde wordt gedaan voor de sensoren aan de achterkant van de auto. Hieermee wordt de versnelling in de 3 richtingen, samen met de yaw en roll rate omgevormd naar te gebruiken waarden. Het resultaat is een ‘YRS’, die alle getransformeerde sensorwaarden bevat om verder in het algoritme te gebruiken.

## 2nd level: Batpack

Alle informatie over het batterijpakket worden hier ingeladen via een CAN bus. Dit bevat de SOC, de temperaturen…

## 2nd level: Drives

Alle informatie over de drives (elektromotoren) worden hier ingeladen via een CAN bus. Informatie over de Torque-current (de stroom die gebruikt wordt om een koppel te genereren), de magnetizing-current , de effectieve snelheid, temperatuur… Het resultaat is 4 verschillende output blokcs, weergegeven door ‘DMCi’ met i gaande van 1 tot 4.

## 2nd level: Dashboard

Dit is alle informatie dat gekozen kan worden door de driver zelf. Zo kan de driver kiezen of hij Torque vectoring algoritme aanzet of niet, bepaalt hij of hij meer aandacht wil geven aan tractiekracht of aan yaw moment, bepaalt hij of tractiecontrole/yaw rate controle al dan niet aanwezig is, bepaalt hij welk percentage van het totaal vermogen gebruikt gaat worden (komt voor als een vermogen limiet op het einde van de controlesystement) en kiest hij hoeveel energie heropgevangen wordt door regeneratieve afremming (moet soms dalen indien de batterijen te warm worden).

## 2nd level: Energy meter

Deze blok neemt informatie op verkregen van de energiemeter en zet deze om in waarden voor de spanning, de stroom… Deze energiemeter zal zeer belangrijk zijn, aangezien de spanning en de stroom die we hier binnenkrijgen wordt vermenigvuldigd om zo het vermogen te verkrijgen dat uit de batterijen wordt gehaald.

# 1st level: Startup Algorithm

Dit element gaat als input alle informatie verkrijgen over de drives, en gaat als uitgangssignaal een vector met 4 elementen, namelijk control, status, buzzer en BMS settings. Deze worden allemaal naar de output blok gestuurd.

## 2nd level: DMC bit extractor

De verschillende signalen van de motor (de status, de stroom…) die opgeslagen werden in een bus, worden nu gesliced om te kijken of de motoren daadwerkelijk klaar zijn om te functioneren. Hierbij maken we gebruik van de bit slicers, die ‘AMK\_status’ gaat ‘slicen’ uit de bus die we als input gaan krijgen, afhangend van de keuze van de MSB (most significant bit) en de LSB (least significant bit). Ik veronderstel dat ‘AMK\_status’ een signaal is bestaande uit een grote hoeveelheid bits (1 en 0) en dat het slicen van deze bits gebruikt kan worden om andere logische schakelingen mee te vormen.

Dit wordt gedaan voor de 4 verschillende motoren, en samen met logische schakelingen wordt er een aantal lokale outputs gegenereerd, zoans ‘DMC\_on’, ‘DMC1\_err’…

## 2nd level: Startup

Een deel van deze outputs (niet de errors en de warning) gaan naar de startup. Dit is in feite een statechart die bekijkt in welke toestand de motoren zich bevinden. Een voorbeeld is de eerste blok, waar een aantal waarden zijn ingevuld voor de verschillende outputs. Indien er aangegeven wordt dat de motoren klaar zijn om te werken (‘DMC\_on ==1’) en dat het stop-signaal uitgeschakeld is, dan kunnen we overgaan naar de volgende toestand. Indien daarna het startsignaal 1 wordt, kunnen de motoren aangezet worden en kan de bestroming beginnen. Het start- en stopsignaal halen we uit het dashboard commando, die de start en stop associeren met een knop voor het aan-of uitzetten. Afhankelijk van de toestand waar we ons bevinden gaan de motoren al dan niet draaien.

## 2nd level: Merge controlword

Afhankelijk van de toestand dat we krijgen, zullen de inputs van deze blok een 1 of een 0 krijgen. Afhankelijk van die waarde wordt er een vector van 5 elementen aangemaakt onder de vorm van ‘control’, die samen met status, buzzer en BMS settings naar de output blok worden gestuurd.

# 1st level: Processor

Hierin gebeurt het eigenlijke werk wat betreft de Torque Vectoring. Als input maakt het gebruik van de sensorwaarden, de toestand/snelheid/bestroming van de drives, de info van het dashboard en de informatie over de energiemeter. Als output geeft het de verschillende koppels die naar de verschillende motoren worden gestuurd samen met de verschillende rotatiesnelheden van de motoren (om bijvoorbeeld het vermogen te limiteren).

## 2nd level: Ackerman Arno

In deze blok wordt er uitgaande van de verplaatsing van uw rack (die veroorzaakt wordt door de rotatie van de steering Wheel) de hoekverdraaiing van de wielen berekend. Dit is een polynomiaalfunctie die afhankelijk is van de geometrie/kinematica van het stuursysteem. Een onderscheid wordt gemaakt tussen het linkse en het rechtse wiel.

## 2nd level: Pedalmapping

In functie van de waarde die de throttle sensor verkrijgt (geschaald om een waarde tussen de 0 en de 100 te verkrijgen) en de waarde van de regen (maximaal vermogen dat we uit het remmen kunnen halen) wordt een effectieve waarde gegeven voor de throttle waarde en voor de brake waarde.

## 2nd level: Lode Torque Conversion block

Alle stromen die gestuurd worden door de ECU naar de drives worden omgezet in overeenstemmende koppels. Dit volgt uit de stroom-koppelkarakteristiek uit de handleiding van de AMK-motoren.

## 2nd level: Sensors

In deze blok wordt gecontroleerd of de sensorwaarden al dan niet plausibele waarden weergeven. Daarom komt er een bepaalde boven- en onderlimiet voor alle sensorwaarden zoals versnelling, steering angle, yaw rate… Hier worden alle sensoren ondergebracht in één vector en wordt een andere vector gedefinieerd voor de berekening van de versnelling van het zwaartepunt.

## 2nd level: Controller Settings

In deze blok maken we de ECU klaar om een rijsetting te kiezen. De inputs worden gegeven door de waarden die op het dashboard worden weergegeven (en die worden ingevoerd door de piloot zelf) en de outputs zijn diezelfde waarden, maar gecomprimeerd in een bus die ‘settings’ heet. Hier wordt er ook gecontroleerd of er in de ‘2nd level: Sensors’ geen errors zijn opgetreden. Indien dit wel het geval is, zal de tractiecontrole afgezet worden en ook de torque vectoring mode zal niet gebruikt worden. We zullen dan op bypass verder rijden.

## 2nd level: Vehicle Estimated Parameters

In deze grote blok gebeurt het eerste belangrijke deel van de berekeningen: Hier worden de verschillende voertuigparameters bepaald en geschat. Een van de belangrijkste bouwblokken is de snelheidsschatting. Daarnaast worden alle krachten bepaald, de waarden van de slip, efficiëntie, de wrijvingsconstante…

### 3th level: Fuzzy velocity

In deze blok gaat de fuzzy-methodologie gebruikt worden om de snelheid van de wagen in te schatten. Hierbij combineren we de informatie van de wielsnelheden (via encoder sensor ingebouwd in de AMK-motor zelf) en de acceleratie van de wagen, die dan geïntegreerd kan worden om zo de snelheid van de wagen te verkrijgen. De fuzzy-methodologie wordt uitgelegd in de thesis.

#### 4th level: Fuzzy parameter

In de Fuzzy parameter blok gaan we de equivalente COG-snelheid van elk wiel bepalen.

#### 4th level: Fuzzy velocity generator

In deze blok maken we gebruik van de voorgaande COG-snelheden en van de gemeten versnellingswaarde om een schatting te maken van de snelheid van het zwaartepunt. Deze zal dan in een feedbackloop geplaatst worden om in de volgende iteratiestap te kunnen gebruiken.

### 3th level: Fz schatting

In deze blok worden de normaalkrachten op elk wiel geschat, uitgaande van een kracht -en momentenevenwicht van de algemene wagen. Een verdeling voor/achter gebeurt via een longitudinaal momentenevenwicht, terwijl de verdeling links/rechts gebruik maakt van een lateraal momentenevenwicht.

### 3th level: Fx schatting

In deze blok worden de tractiekrachten (longitudinale) krachten op de wielen geschat. Dit doen we op basis van het koppel dat uitgeoefend wordt op het wiel (eventueel inertie van de motor en de banden in rekening houden), maar kan efficiënter gedaan worden indien we werken met bandenmodellen, zoals Magic Tyre Formula.

### 3th level: Fy schatting

In deze blok worden de laterale krachten op de wielen geschat. Dit doen we door te kijken naar de 2 bijdragen die aanwezig zijn voor het yaw moment: de longitudinale en de laterale krachten. Beiden hebben een bijdrage. Aangezien F\_x op elk wiel al berekend is in de vorige stap, kan het yaw moment ten gevolge van de laterale krachten gehaald worden uit het totale yaw moment (gegeven door de inertie\*hoekversnelling van de wagen rond de z-as. Maar opnieuw kan dit efficiënter gedaan worden indien we werken met bandenmodellen, zoals Magic Tyre Formula.

### 3th level: Yaw rate reference schatting

In deze blok gaan we de referentie yaw rate gaan definiëren, die zeer belangrijk zal zijn bij de controle in het TV algoritme. Op deze waarde zal de werkelijke yaw rate immers gecorrigeerd worden. We maken hier gebruik van de Ackermann neutral steering veronderstelling voor de yaw rate.

### 3th level: Aero schatting

Hier gaan we, uitgaande van de snelheid die geschat werd, de drag en lift krachten berekenen die gaan inwerken op de auto. Hier kunnen eventueel aanpassingen gedaan worden om de voor- en achtervleugel apart te beschouwen, eventueel ook rekening houden met een actief aero-pakket vooraan of achteraan… Indien er actief gewerkt wordt, zal binnen de aero schatting ook een controle-algoritme geschreven dienen te worden.

### 3th level: Mu/slip schatting

In deze blok gaan we de wrijvingscoëfficiënt bepalen tussen de banden en het wegdek en ook gaan kijken naar de waarden van de slip op elk wiel. Voor het berekenen van de slip hebben we de longitudinale component van de snelheid nodig, samen met de rotatiesnelheid van de wielen (te bepalen aan de hand van rotatiesnelheid van de motor).

### 3th level: Efficiency schatting

In deze blok gaan we proberen om de efficiëntie zo goed mogelijk te controleren, om ervoor te zorgen dat het mechanisch vermogen zo accuraat mogelijk kan berekend worden en dat zo het maximaal elektrisch vermogen nooit overschrijd wordt.

## 2nd level: Control Systems

Dit is de belangrijkste blok van het hele Torque Vectoring algoritme. Hier worden als input de verschillende sensorwaarden, de schattingen van snelheid, kracht… en de settings ingeladen. Eerst wordt de yaw rate gecontroleerd. Daarna wordt er een torque distribution algoritme gebruikt om op een zo efficiënt mogelijke manier de koppels te verdelen over de verschillende motoren. Daarna wordt er gekeken naar de Traciecontrole en uiteindelijk wordt nagegaan of het maximaal vermogen niet overschreden wordt.

### 3th level: Yaw rate control

In deze blok zal de yaw rate gecontroleerd worden. Vertrekkende van de referentiewaarde van de yaw rate, gaan we deze vergelijken met de werkelijke waarde die via de sensoren wordt opgemeten. Deze error wordt dan doorgestuurd naar een PID controller. Na de optelling van de proportionele/ integrale/derivatieve term (waarbij ook de invloed van de verandering in steering angle en de verandering van het yaw moment in rekening wordt gebracht), tellen we nog een extra term bij, proportioneel aan de steering angle. Dit is de referentiewaarde voor het moment M\_z. Deze gaat dan naar het TV algoritme.

### 3th level: Torque distributor

In deze blok wordt de verdeling van de koppels over de 4 verschillende motoren gerealiseerd. De output is een koppel naar elke motor.

#### 4th level: Torque/brakes analyser

Hier worden de desired versnellingskrachten en remkrachten berekend op basis van de waarde van de throttle en de maximale kracht die de auto longitudinaal zou kunnen uitoefenen, uitgaande van de totale verticale kracht die inwerkt op de banden.

#### 4th level: Torque distributor

In deze blokken worden de koppels effectief verdeeld over de 4 wielen. Dit gebeurt in functie van de stand waarin de auto zich bevindt (of de auto nu wel of niet het TV algoritme zal toepassen). Indien de auto geen TV gebruikt, zal er een even grote hoeveelheid torque aan de voor-en achterwielen ontrokken (binnenkant bocht) of opgeteld (buitenkant bocht) worden.

### 3th level: Traction Control

In deze blok wordt de tractiecontrole uitgeschreven. Door middel van een PID controller wordt de waarde van de slip gecontroleerd rond een optimale waarde van ongeveer 0.1. Uitgaande van de fout op de slip wordt het koppel aangepast aan elk wiel, NADAT de torque distribution gebeurt.

### 3th level: Power checker

In deze blok controleren we of de limiet van het elektrisch vermogen niet wordt overschreden. Indien het mechanisch vermogen de maximale waarde overschrijdt, wordt het koppel verminderd in functie van het verschil in vermogen.