

Case Simulink

EE4- Building a SSV - Team PM1
21 maart 2014



Inhoudsopgave

Inhoudsopgave	1
Figurenlijst	1
Inleiding	2
Gedrag van het zonnepaneel gekoppeld aan een weerstand	2
Gedrag van de DC-motor zonder zonnepaneel.....	5
Racesimulatie	8
Vergelijking MATLAB-simulatie en analytische model.....	11

Figurenlijst

Figuur 1- Simulinkmodel zonnepaneel.....	2
Figuur 2- Spanningsstroomkarakteristieken ('trendlijn' en gesimuleerde waarden)	3
Figuur 3- Spannings-vermogenkarakteristieken ('trendlijn' en gesimuleerde waarden)	3
Figuur 4- Paramaters Simulink (zonnepaneel)	4
Figuur 5-Situatieschets: stoot van bal tegen wagen	5
Figuur 6-Model van de motor gebruikt voor de simulatie.....	6
Figuur 7-Grafiek uit Simulink: snelheid in functie van de tijd	7
Figuur 8-Grafiek uit Simulink: afgelegde weg in functie van de tijd	7

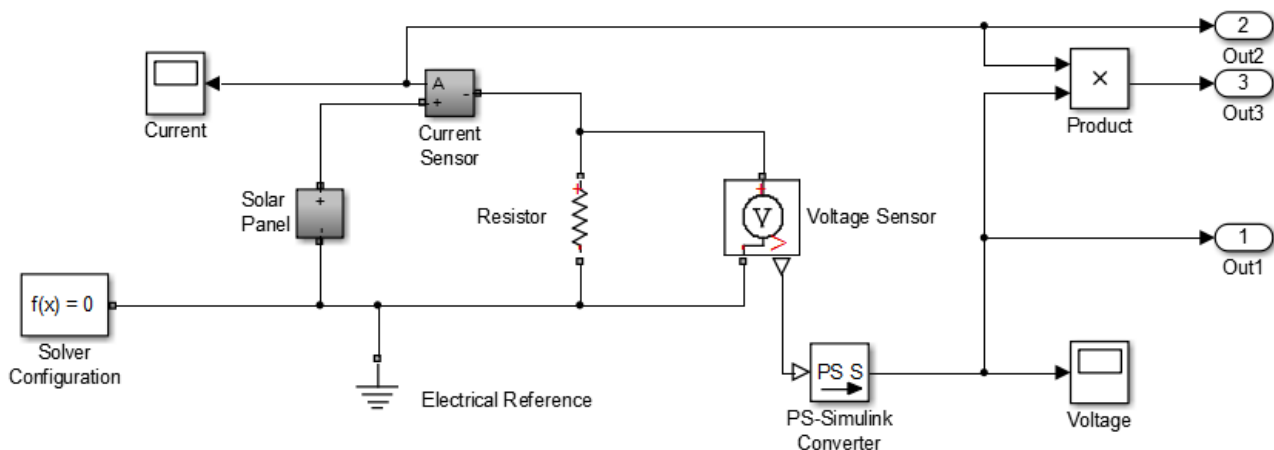
Inleiding

In deze case is het de bedoeling om via Matlab en Simulink het gedrag van onze SSV te simuleren. Dit zal verwezenlijkt worden door eerst verschillende parameters afzonderlijk te testen, zoals het gedrag van ons zonnepaneel als het enkel aan een weerstand gekoppeld is.

De bedoeling van al deze simulaties is om onze wagen gedurende het project aan te passen aan de hand van de resultaten van de simulaties. Dit zal uiteindelijk leiden tot een optimaal werkend wagentje.

Gedrag van het zonnepaneel gekoppeld aan een weerstand

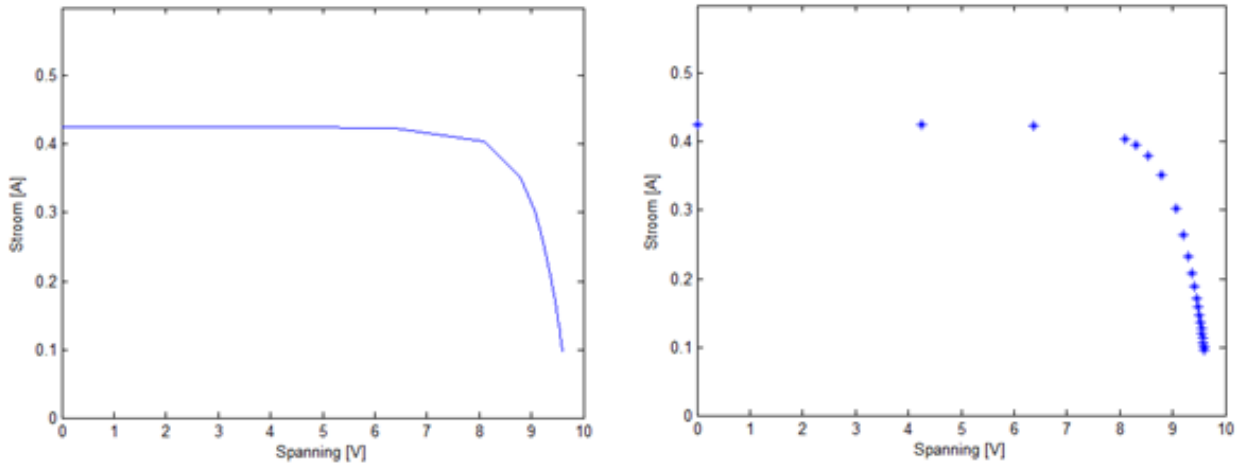
Bij deze simulaties was het de bedoeling om verschillende weerstanden, gaande van 10Ω tot 100Ω (met betekenisvolle stappen), aan een model van het zonnepaneel te hangen. Dit model was reeds voor ons gemaakt en konden we vinden op Toledo.



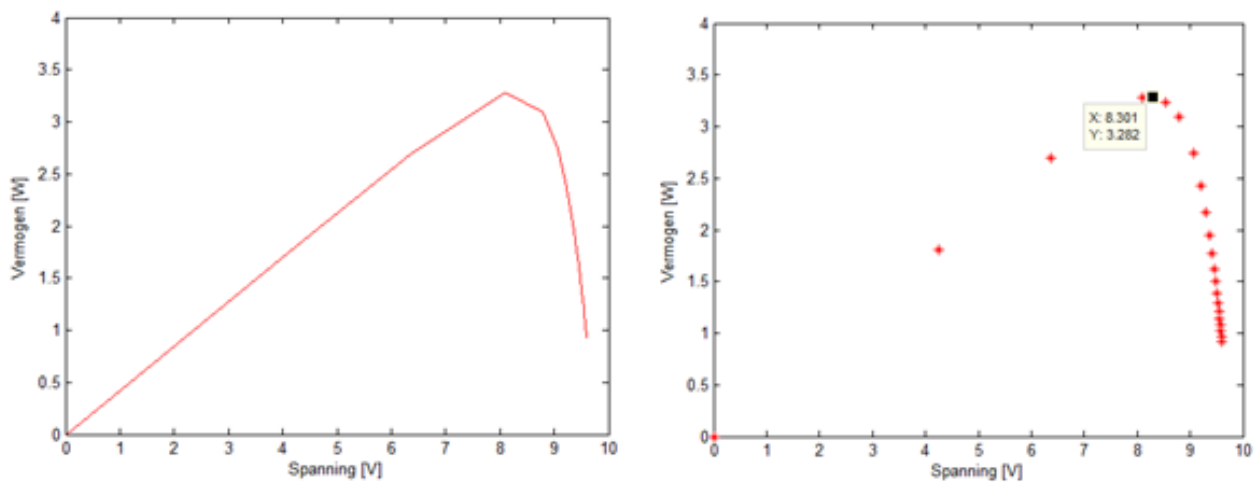
Figuur 1- Simulinkmodel zonnepaneel

Via deze simulaties moet de weerstand bepaald worden waarbij het geleverd vermogen maximaal is. Als we de grafieken (zoals hieronder) laten tekenen met enkel de punten in de plaats van de volledige grafiek komen we uit bij een maximaal vermogen voor een weerstand van 21Ω .

In de grafieken die hieronder getekend zijn hebben we de weerstandswaarde laten beginnen vanaf 0Ω zodat we een mooie grafiek bekwamen die vertrekt vanuit de oorsprong.



Figuur 2- Spanningsstroomkarakteristieken ('trendlijn' en gesimuleerde waarden)



Figuur 3- Spannings-vermogenkarakteristieken ('trendlijn' en gesimuleerde waarden)

Deze simulaties hebben we uitgevoerd aan de hand van de m-file die voor ons ter beschikking werd gesteld op Toledo. Hierin moesten we zelf de parameters (zoals de weerstanden) invullen voordat we de simulaties konden laten lopen. Alle parameters die we hebben ingevuld zijn te vinden in figuur 4.

```

clear all;
close all;

%%% Solar Power
Ir = 850 ;
Is = 1e-8 ;
Isc = 0.35 ;
Voc = 9.63 ;
Ir0 = 700 ; % irradiance used for measurements [W/m^2]
m = 20.96 ;
% if you use only 1 cell, you have to multiply your diode factor with the
% actual number of cells on the solar panel

V=[];
I=[];
P=[];

% replace these values for the resistance with relevant values
R_list=[0 10 15 20 21 22.5 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100];
for i=1:length(R_list)
    R=R_list(i)

    sim('Solar_panel_model',1); % simulate Simulink model "Solar_panel_model.mdl" for 1 s

    V = [V yout(end,1)];
    I = [I yout(end,2)];
    P = [P yout(end,3)];
end

figure(1)
plot(V,I,'b*');
ylabel('Stroom [A]');
xlabel('Spanning [V]');
axis([0 10 0 0.6]);
set(gcf,'color',[1 1 1])

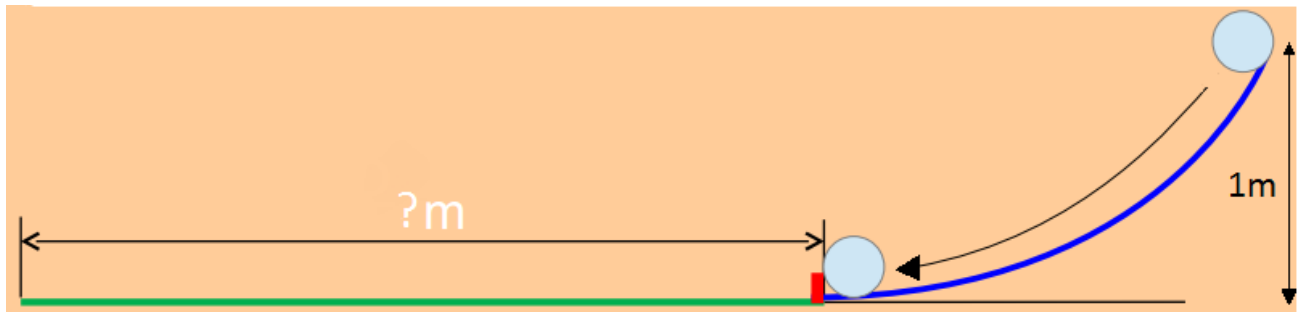
figure(2)
plot(V,P,'r*');
ylabel('Vermogen [W]');
xlabel('Spanning [V]');
axis([0 10 0 4]);
set(gcf,'color',[1 1 1])

```

Figuur 4- Paramaters Simulink (zonnepaneel)

Gedrag van de DC-motor zonder zonnepaneel

Bij dit onderdeel is het de bedoeling dat we simuleren hoe ver onze SSV zal rijden nadat hij weggeduwd wordt door de petanquebal die vertrekt op een hoogte van 1 meter. Bij deze simulaties zullen we geen gebruik maken van het zonnepaneel.



Figuur 5-Situatieschets: stoot van bal tegen wagen

Voordat we de simulatie kunnen uitvoeren moeten we eerst de snelheid berekenen die de bal aan ons wagentje zal geven vlak na de botsing, die we ideaal veronderstellen. Deze berekeningen zullen we doen aan de hand van de kinetische energie tijdens de botsing en de potentiële energie van de bal.

$$v_{end} = \sqrt{2gh} \quad \text{en} \quad v_{end} = \frac{2M_{ini,SSV}v_{ini,SSV}}{M_{ini,SSV} + M_{ini,ball}}$$

Daaruit volgt:
$$\sqrt{2gh} = \frac{2M_{ini,SSV}v_{ini,SSV}}{M_{ini,SSV} + M_{ini,ball}}$$

Hierbij is $g=9,81\text{m/s}^2$; $h=1\text{m}$; $M_{ini,SSV}=1,690\text{kg}$; $M_{ini,ball}=0,735\text{kg}$

$$v_{ini,SSV} = \frac{(M_{ini,SSV} + M_{ini,ball})\sqrt{2gh}}{2M_{ini,SSV}} = 3,18\text{m/s}$$

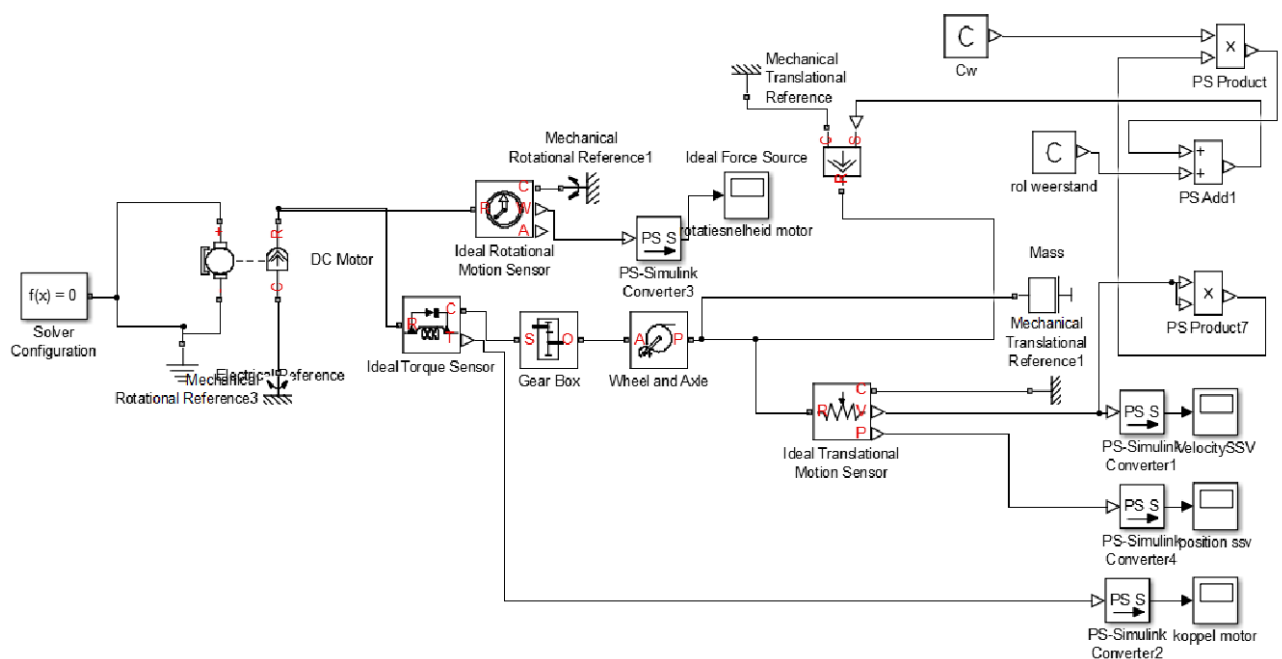
Aangezien we enkel het model van de motor gebruiken moeten we ook een snelheid van deze motor kennen. Deze snelheid zullen we gelijkstellen aan de snelheid die de wagen zou krijgen bij de botsing. We zullen dus eigenlijk geen botsing simuleren, maar wel het ‘uitbollen’ van onze wagen voor een bepaalde beginsnelheid. Deze beginsnelheid kunnen we echter niet ingeven in de eenheid m/s maar deze moet worden ingegeven als een toerental. Dit toerental bekomen we door de snelheid te delen door de straal.

Opdat ons wagentje zou vertragen en uiteindelijk tot stilstand zal komen kennen we ook een last-koppel toe aan onze motor. Dit lastkoppel baseren op volgende functies:

$$F_w = \frac{1}{2}C_w A \rho v^2$$

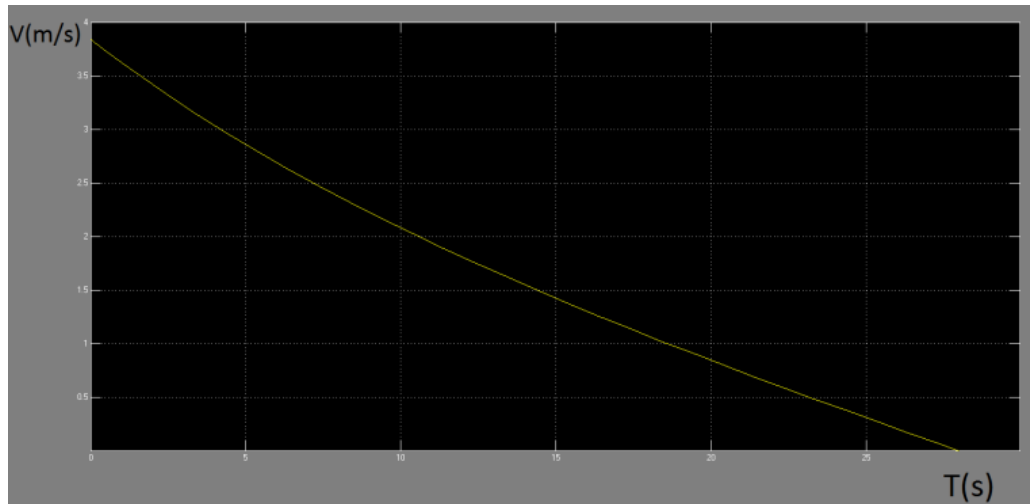
$$F_r = C_{rr} N$$

Dit is nodig doordat er geen component die een koppel kan genereren, namelijk het zonnepaneel, aan de motor gekoppeld is.



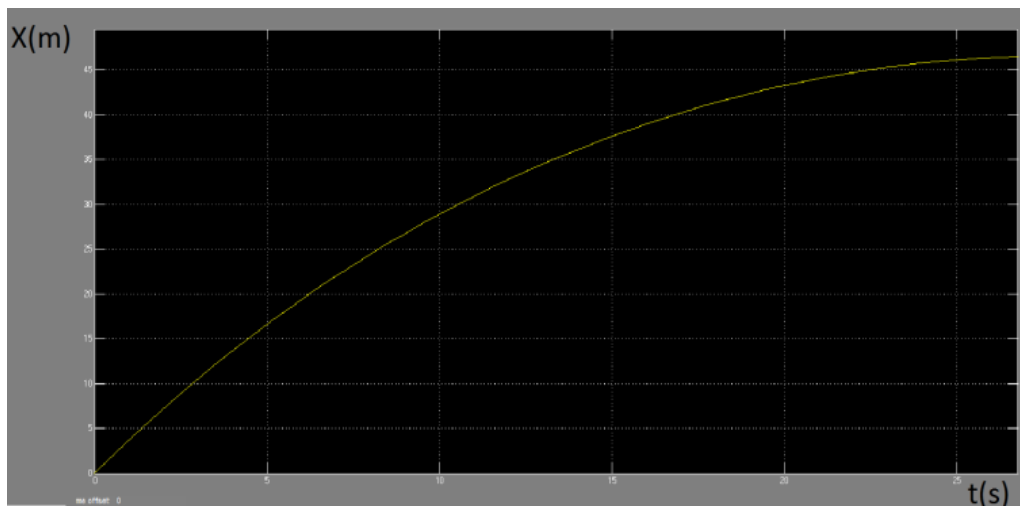
Figuur 6-Model van de motor gebruikt voor de simulatie

Als de simulatie wordt uitgevoerd worden er twee verschillende grafieken getekend, één grafiek is de snelheid in functie van de tijd en de andere grafiek is de positie in functie van de tijd. Voor de positie zijn we uitgegaan van een startpositie gelijk aan 0 meter. Als we de grafiek van de snelheid analyseren zien we dat de snelheid sterk daalt. De reden hiervoor is dat de luchtweerstand snelheidsafhankelijk is. Deze relatie is kwadratisch, dit wil zeggen dat wanneer de snelheid hoog is, de kracht groot zal zijn. Nadat de snelheid gedaald is, na ongeveer 15 seconden, zien we dat de snelheid lineair daalt. Dit komt doordat er bij lage snelheden vooral rolwrijving is, deze rolwrijving is eigenlijk een constante wrijving waardoor er sprake is van een lineair verband.



Figuur 7-Grafiek uit Simulink: snelheid in functie van de tijd

Op bovenstaande grafiek kunnen we ook nog zien dat de wagen tot stilstand komt na ± 28 seconden. Als we de grafiek van de afgelegde weg in functie van de tijd gaan analyseren zien we dat ons wagentje in het eerste tijdsinterval, van 0 tot ongeveer 15 seconden, het grootste deel van de weg heeft afgelegd. Vanaf 15 seconden, het moment dat de snelheid lineair verder daalt, zien we dat de wagen nog maar heel weinig afstand aflegt.

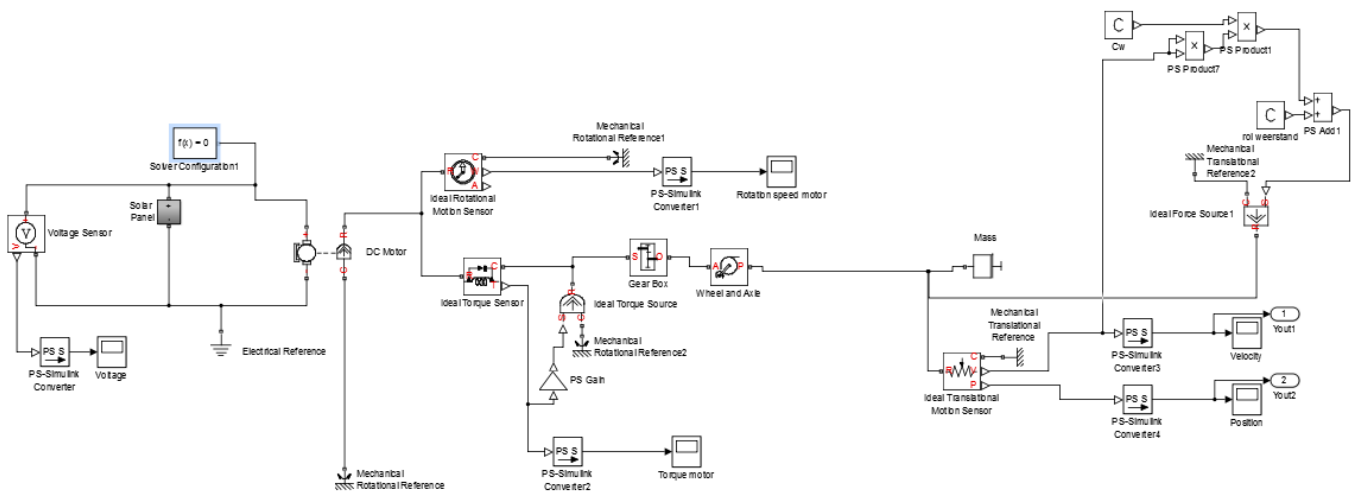


Figuur 8-Grafiek uit Simulink: afgelegde weg in functie van de tijd

Racesimulatie

Bij dit deel van case simulink is het de bedoeling om de race zelf te gaan simuleren of nabootsen. Het doel van deze simulatie is het vinden van een gear ratio en een optimale massa. Ons schema is opgebouwd uit een zonnepaneel, een motor en een last of weerstand; deze is toegevoegd om zo goed als mogelijk de realiteit te simuleren. Deze weerstand omvat de luchtweerstand en de rolweerstand. We werken volgens dezelfde formules die we ook vorige keer hebben gebruikt.

In tegenstelling tot vorige vraag gaan we nu wel een stroom en een spanning over de motor plaatsen door middel van het zonnepaneel.



Figuur 9-Simulinkmodel race

Aangezien we ook al via MATLAB simulaties hebben uitgevoerd, hebben we al een goed idee van de optimale massa en gear ratio. Op basis daarvan kunnen we ons nu beperken tot simulaties met massa's en gear ratios die rond de eerder bekomen waarden liggen.

```

M_ball = 0.735; % kg
% the two following variables are vectors, which you can define as follows:
% example_list = [500 520 540 560]; %or% example_list = (500:20:560);
M_list = [1.4 1.45 1.5 1.55] ; % kg
ratio_list = [11 12 13 14];%

for i=1:length(M_list)
    mass = M_list(i) % iterate through all

    for j=1:length(ratio_list)

        ratio = ratio_list(j) % for each mass, iterate through all ratios
        % Run simulation for 10 s
        sim('SSV',45);

        index=find(yout(:,2)>=10,1);
        speed(i,j) = yout(index-1,1)+(yout(index,1)-yout(index-1,1))*(10-yout(index-1,2))/(yout(index,2)-yout(index-1,2))

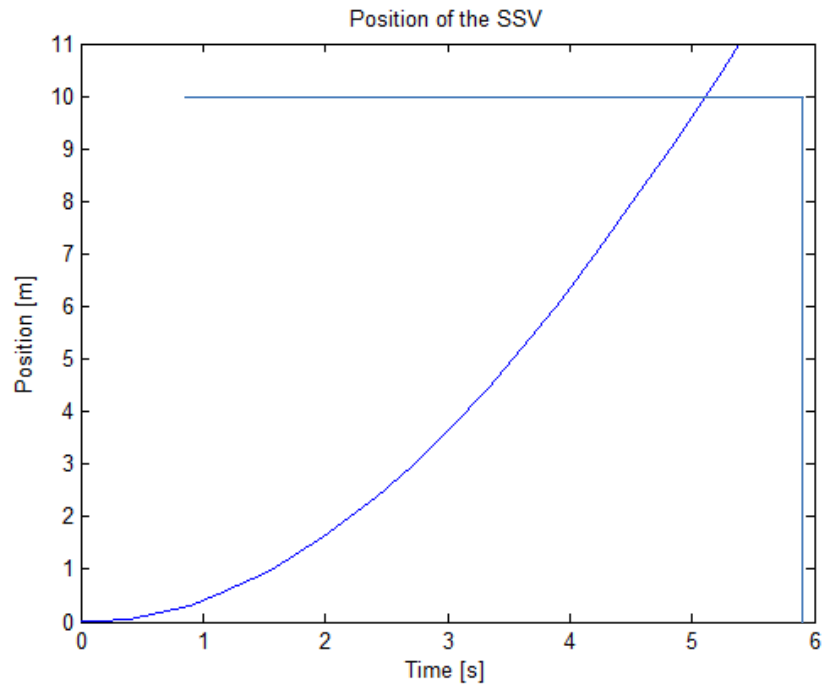
        % Here you might want to calculate the goal function.
        speedball(i,j) = (2*mass*speed(i,j))/(mass + M_ball)
    end
end

```

Figuur 10-Verschillende Gear Ratio's en massa's ingevoerd in Simulink + de functie om de snelheid van de bal te berekenen

Zo zal MATLAB voor alle massa's simuleren met alle gegeven gear ratio's. Op het einde van de berekeningen zal het weergeven welke de optimale massa en gear ratio zijn respectievelijk 1.45kg en 12. Dit komt overeen met wat we vonden uit de bisectie methode

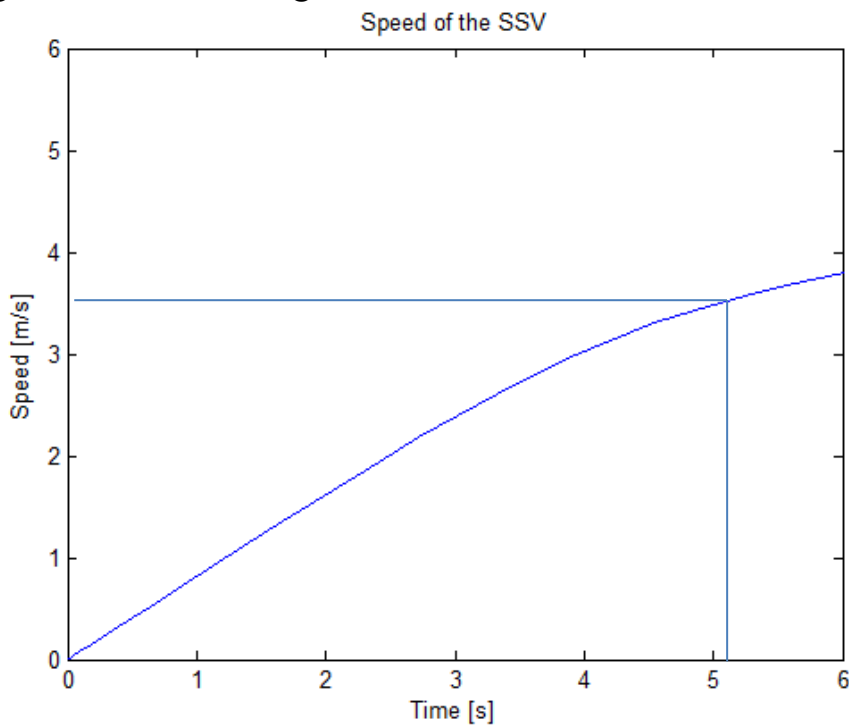
We vragen het ook om d.m.v. een plot functie de snelheid en de positie grafisch weer te geven. Op deze manier kunnen we dus ook grafisch zien hoe de SSV zal rijden en welke afstand hij heeft afgelegd. Ook kunnen we de snelheid op verschillende tijdstippen aflezen.



Figuur 11-Positie in functie van tijd

Op deze grafiek kan men ook aflezen wat de verstreken tijd is nadat de SSV 10 meter heeft afgelegd. Deze zal er dus iets meer dan 5 seconden over doen om de baan volledig af te leggen.

Om nu de snelheid te kunnen bepalen op het einde van de baan maken we gebruik van de t,v-grafiek.



Figuur 12-Snelheid in functie van tijd

Op die manier kunnen we afleiden dat de snelheid van de SSV bij botsing met de bal een waarde van ongeveer 3.6m/s zal bereiken.

Vergelijking MATLAB-simulatie en analytisch model

Door de verschillende situaties, die hierboven uitgebreid beschreven zijn, te simuleren in MATLAB hebben we een beter beeld gekregen over de werking van onze kleine zonnwagen. Zo konden we zien dat de rol- en luchtweerstand wel degelijk een grote invloed hebben op de afstand die onze wagen zou afleggen. Deze invloeden werden niet in rekening gebracht in ons analytisch model waardoor we uitkwamen op redelijk grote verschillen. Zo werd de massa van ons wagentje plots met 220 gram verhoogt. Deze informatie zorgt ervoor dat het design van ons wagentje kan worden aangepast alvorens het gemaakt is. Dit heeft als gevolg dat er veel tijd bespaard wordt in het aanpassen van het werkelijke model.