



MIDNIGHT SUN SOLAR CAR TEAM
UNIVERSITY OF WATERLOO

FEA Boundary Conditions MSXII

Prepared by:
Devon Copeland

September 28, 2017

1 Background

1.1 Vehicle Design

Midnight Sun Twelve (MSXII) is a cruiser class, solar electric vehicle being designed with the goal of competing in the 2018 American Solar Challenge (ASC 2018) and the 2019 World Solar Challenge (WSC 2019). By definition, cruiser class solar vehicles must be multi-occupant and are designed with the intent of being more practical than a typical, challenger class solar car. Because of the unique requirements of this class, the spring rates and target damping coefficients on MSXII's suspension must be selected to optimize for efficiency while still keeping driver comfort in mind. This report proposes an analytical technique for determining the response of a suspension system to any given road conditions.

1.2 Important Vehicle Parameters

Table 1 lists vehicle parameters that are of importance to the design and analysis of MSXII's suspension. The mass and moment of inertia was computed by CAD software.

Table 1: Important Vehicle Parameters

Mass (M)	550	kg
Wheelbase	2.60	m
Track	1.60	m
Tire Diameter	0.53	m
Wheel Assembly Moment of Inertia (J_w)	0.25	kgm^2
Distance from Front Wheel to CoG (a)	1.32	m
Distance from Road to CoG (h)	0.52	m
Full Wheel Travel in Compression	45	mm
Front Coilover Angle from Vertical	40	$^\circ$
Rear Coilover Angle from Vertical	30	$^\circ$

1.3 Suspension Architecture

MSXII's front suspension comprises of a double wishbone linkage with an outboard coilover while the rear suspension comprises of independent trailing arms allowing for zero scrub and thus less rolling resistance. Figure 1 shows the screenshots of the front and rear suspension.

1.4 Approach to FEA

explain the different loading conditions to be tested (dynamic, static, fatigue etc.)

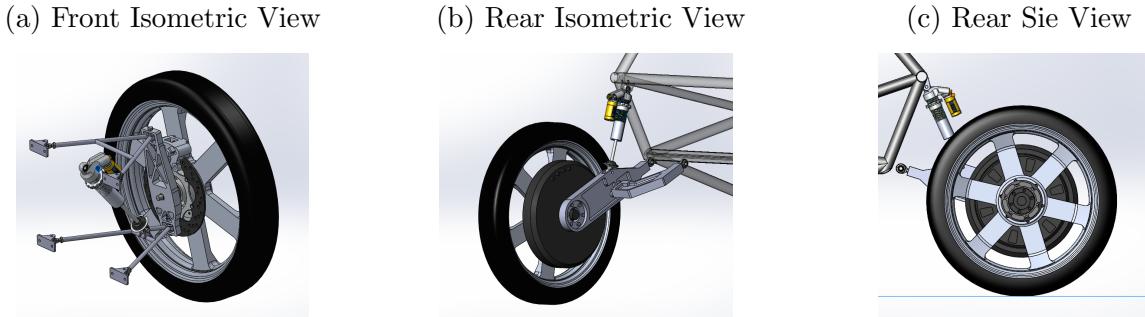


Figure 1: MSXII Suspension Design (Screenshots from Solidworks)

2 Static Loading

Assuming a smooth driving surface, the largest normal force acting on the front tire will occur while braking in a turn. Similarly, the largest normal force acting on the rear tire will occur while accelerating in a turn. To approximate these loading condition, the a superposition of two static half car models is used; one for cornering and one for hard braking.

2.1 Hard Braking

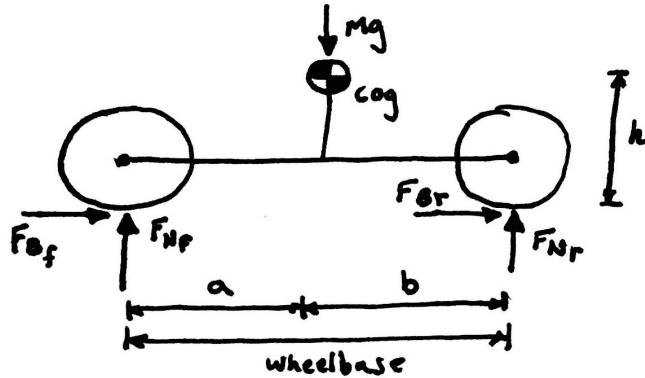


Figure 2: Free Body Diagram of a Car Braking

Figure 2 shows a free body diagram of a half car model undergoing braking. MSXII will only have brake callipers on the front wheels however regenerative braking from the hub motors will also provide a braking force at the rear contact patches. Assuming a generous coefficient of static friction, μ_s , of 1.0 and the extreme case where the both the font and rear tires are about to begin sliding, the combined braking force can be expressed as:

$$F_{braking} = F_{Nf} + F_{Nr} = \mu_s F_{N_{net}} = \mu_s Mg = (1.0)(550\text{kg}) \left(9.81 \frac{\text{m}}{\text{s}^2} \right) = 5.40\text{kN} \quad (1)$$

From Figure 2, summing the moments about the centre of gravity and the vertical forces results in the following equations:

$$aF_{Nf} = hF_{braking} + bF_{Nr} \quad (2)$$

$$F_{Nf} + F_{Nr} = Mg \quad (3)$$

Solving for the front normal force:

$$\begin{aligned} aF_{Nf} &= hF_{braking} + b(Mg - F_{Nf}) \\ F_{Nf} &= \frac{hF_{braking} + bMg}{a + b} = \frac{(0.52m)(5.40kN) + (1.28m)(550kg)(9.81\frac{m}{s^2})}{(1.32m) + (1.28m)} = 3.74kN \end{aligned} \quad (4)$$

Note that 3.74kN is for both front wheels. Therefore there is approximately **1.87kN** of normal force on each front tire during hard braking.

2.2 Maximum acceleration

MSXII will use SCM-150s which have a peak torque of 135Nm, however from the manufacturer provided torque speed curves, a more reasonable upper limit to for peak torque sustainable as the car accelerates is approximately 100Nm [2]. From this torque value and a number of other vehicle parameters the vehicle's maximum acceleration can be calculated as follows:

$$a = \frac{F_w}{\frac{1}{2}M} \quad (5)$$

$$\alpha_w = \frac{T_m - F_w r}{J_w} \quad (6)$$

$$\alpha_w = ar \quad (7)$$

where a is the acceleration of the vehicle, F_w is the friction force acting on a single tire while accelerating, J_w is the moment of inertia of the wheel assembly, r is the radius of the tire, T_m is the torque from the motor and α_w is the angular acceleration of the wheel.

Rearranging the above three equations, the acceleration of the vehicle can be expressed as a function of motor torque as follows:

$$\begin{aligned} a &= \frac{T_m r}{\frac{1}{2}Mr^2 + J_w} \\ a &= \frac{(100Nm)(0.27m)}{\frac{1}{2}(550kg)(0.27m^2)^2 + 0.25kgm^2} = 1.33m/s^2 \end{aligned} \quad (8)$$

Solving for friction force on a single rear tire:

$$F_w = Ma = (550kg)(1.33m/s^2) = 732N \quad (9)$$

Using the same approach as demonstrated in section 2.1, the resulting normal force on the rear wheels can be calculated as follows:

$$\begin{aligned} bF_{Nr} &= hF_w + a(Mg - F_{Nr}) \\ F_{Nr} &= \frac{hF_w + aMg}{a + b} = \frac{(0.52m)(732N) + (1.32m)(550kg)(9.81\frac{m}{s^2})}{(1.32m) + (1.28m)} = 2.89kN \end{aligned} \quad (10)$$

Note that 2.88kN is for both rear wheels. Therefore there is approximately **1.44kN** of normal force on each rear tire during maximum acceleration.

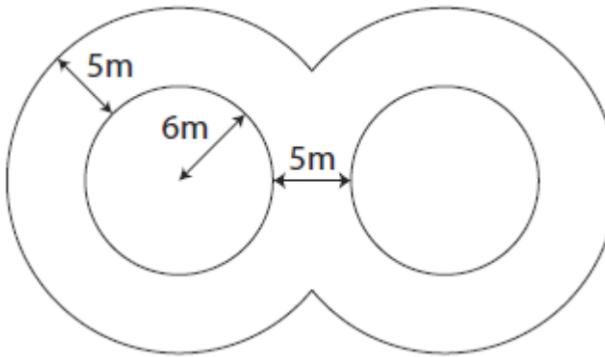


Figure 3: Figure of Eight Course from ASC 2018 Regulations [1]

2.2.1 Cornering

To determine the lateral force subjected to the vehicle, the worst case of two scenarios is selected. The first scenario originates from the ASC 2018 regulations while the second draws its roots from actual highway geometry and speed limits.

For the first scenario, the ASC 2018 regulations stipulate that a vehicle must be able to navigate a figure of eight as shown in Figure 3 in less than 18 seconds [1]. Since the average arc length of the figure of eight is $34\pi m$, the net lateral force required to navigate the figure of eight can be found as follows:

$$F_{si} + F_{so} = F_{steer} = \frac{Mv^2}{r} = \frac{(550kg) \left(\frac{34\pi m}{18s}\right)^2}{8.5m} = 2.28kN \quad (11)$$

For the second scenario, the diameter of several highway exit ramps around the University of Waterloo were measured as shown in Figure 4. The recommended speed limit on these exits is $50km/hr$. Assuming a worst case scenario where the vehicle takes an exit with a $100m$ diameter at $60km/hr$ ($16.67m/s$), required lateral force required to navigate this corner can be found as follows:

$$F_{steer} = \frac{Mv^2}{r} = \frac{(550kg)(16.67m/s)^2}{50m} = 3.06kN \quad (12)$$

Since the lateral force required in the second scenario is highest, $3.06kN$ will be used as the worst case cornering force.

Using the half car model shown in Figure 5, summing the moments about the centre of gravity and the vertical forces results in the following equations:

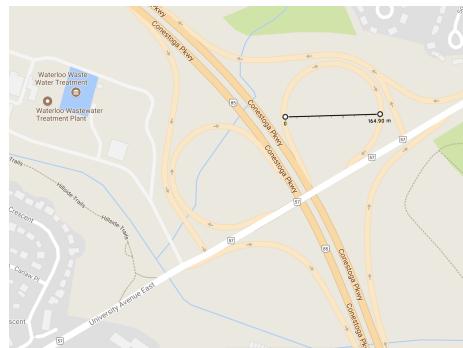
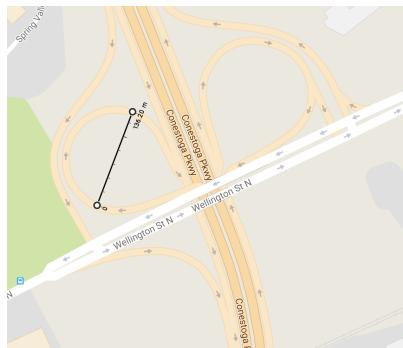
$$\frac{\text{track}}{2} F_{No} = hF_{steer} + \frac{\text{track}}{2} F_{Ni} \quad (13)$$

$$F_{No} + F_{Ni} = Mg \quad (14)$$

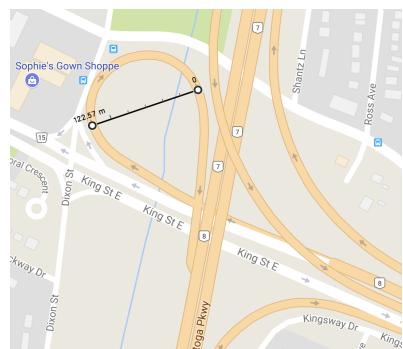
Solving for the outside wheel normal force:

$$\begin{aligned} \frac{\text{track}}{2} F_{No} &= hF_{steer} + \frac{\text{track}}{2} (Mg - F_{Ni}) \\ F_{No} &= \frac{hF_{steer} + \frac{\text{track}}{2} Mg}{\text{track}} = \frac{(0.52m)(3.06kN) + (0.80m)(550kg) \left(9.81 \frac{m}{s^2}\right)}{1.60m} = 3.69kN \end{aligned} \quad (15)$$

(a) Conestoga Pkwy. & Wellington St. (b) Conestoga Pkwy. & University Ave.



(c) Conestoga Pkwy. & King St.



(d) Speed Limit on Conestoga Pkwy Exits



Figure 4: Highway Exist Around the University of Waterloo

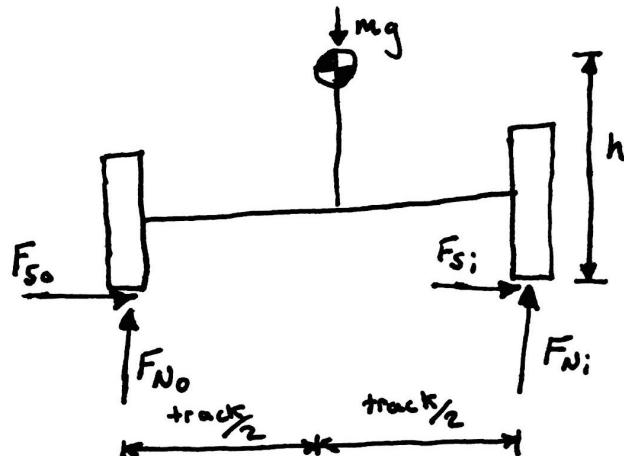


Figure 5: Free Body Diagram of a Car Cornering

Solving for the inside wheel normal force:

$$F_{Ni} = Mg - F_{No} = (550\text{kg}) \left(9.81 \frac{\text{m}}{\text{s}^2} \right) - 3.69\text{kN} = 1.71\text{kN} \quad (16)$$

Note that the forces above are for both front and rear outside wheels. Assuming the force is evenly split between the front and rear, there is approximately **1.85kN** of normal force on each outside tire and **0.85kN** for each inside tire during maximum cornering.

2.3 Superposition

The braking accelerating and cornering conditions subject the wheels to normal forces that differ from those seen when the car is at rest. The change in normal force for each loading condition can be expressed as follows:

$$\begin{aligned}\Delta F_{Nf_{braking}} &= F_{Nf_{braking}} - \frac{Mg}{4} = 1.87kN - 1.35kN = 0.52kN \\ \Delta F_{Nr_{accelerating}} &= F_{Nr_{accelerating}} - \frac{Mg}{4} = 1.44kN - 1.35kN = 0.09kN \\ \Delta F_{No_{cornering}} &= F_{No_{cornering}} - \frac{Mg}{4} = 1.85kN - 1.35kN = 0.50kN \\ \Delta F_{Ni_{cornering}} &= F_{Ni_{cornering}} - \frac{Mg}{4} = 0.85kN - 1.35kN = -0.50kN\end{aligned}\tag{17}$$

To approximate the worst case loading condition, a superposition of the changes in normal force and the nominal normal force is applied. Note that the nominal force used is the same for each wheel which assumes a centre of gravity approximately centred along the wheelbase.

$$\begin{aligned}F_{Nfi_{max}} &= \Delta F_{Nf_{braking}} + \Delta F_{Ni_{cornering}} + F_{N_{nominal}} = 0.52kN - 0.50kN + 1.35kN = 1.24kN \\ F_{Nfo_{max}} &= \Delta F_{Nr_{braking}} + \Delta F_{No_{cornering}} + F_{N_{nominal}} = 0.52kN + 0.50kN + 1.35kN = 2.37kN \\ F_{Nri_{max}} &= \Delta F_{Nr_{braking}} + \Delta F_{Ni_{cornering}} + F_{N_{nominal}} = 0.09kN - 0.50kN + 1.35kN = 0.94kN \\ F_{Nro_{max}} &= \Delta F_{Nr_{braking}} + \Delta F_{No_{cornering}} + F_{N_{nominal}} = 0.09kN + 0.50kN + 1.35kN = 1.94kN\end{aligned}\tag{18}$$

2.4 Summary

Table ?? summarizes the loading conditions used for static structural FEA simulations on the suspension and steering components. Note that the direction of the force corresponds with the global coordinate system of the MSXII and the vectors shown in Figure 6. The assemblies shown are for the driver's side of the vehicle

Table 2: Summary of Suspension Loading Conditions

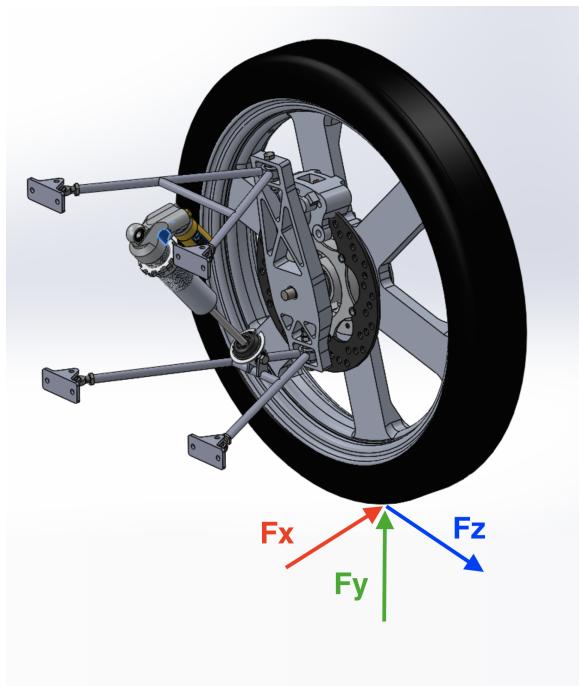
Case	Fx (kN)	Fy (kN)	Fz (kN)
Braking & Cornering (Inside Front Wheel)	1.53	1.24	2.70
Braking & Cornering (Outside Front Wheel)	-1.53	2.37	2.70
Accelerating & Cornering (Inside Rear Wheel)	1.53	0.94	0.73
Accelerating & Cornering (Outside Rear Wheel)	-1.53	1.94	0.73

Since the lateral force on each of the wheels in particular is dependent on a number of tire properties that are difficult to model (slip angle, tire pressure, etc.). It was assumed that 50% of the net lateral force will be the maximum lateral force experienced by any one of the four wheels at a given time.

3 Dynamic Loading

MSXII will be using Ohlins TTX25 dampers for both the front and rear suspension. From the Ohlins' published data, [3], the maximum damping coefficient achievable is approximately

(a) Front Suspension



(b) Rear Suspension

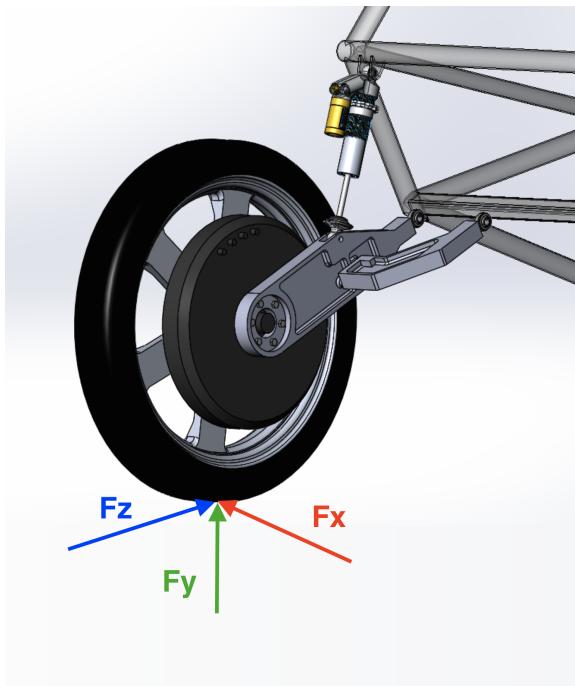


Figure 6: Summary of Suspension Loading Conditions

3.5kNs/m.

References

- [1] American Solar Challenge. 2018 regulations, [online], 2017. <http://americansolarchallenge.org/ASC/wp-content/uploads/2016/08/ASC2018-Regs-External-Revision-A.pdf>.
- [2] New Generation Motors Corp. Understanding your dampers: A guide from jim kasprzak, [online], 2002. http://vignette1.wikia.nocookie.net/solarcar/images/c/c3/NGM_SC-M150_Specs.pdf/revision/latest?cb=20100803000620.
- [3] Ohlins USA. Ttx25 mkii dyno plots, [online], 2015. <https://www.ohlinsusa.com/suspension-products/Automotive/Auto>

A MATLAB Source Code for Simplified Model

```

1 % ----- %
2 %
3 % Suspension Spring Rates Project - alternativeFormulation.m
4 % Function to setup transfer functions for a half car model to determine the
5 % response of MSXII to disturbances in the road conditions
6 %
7 % Author: Devon Copeland
8 %
9 % Midnight Sun 2017
10 %
11 %
12 % Notes:
13 %
14 % -
15 % ----- %
16
17 load_system('halfCarModel');
18 close all;
19 clear all
20
21 % ----- Constants ----- %
22
23 c1_ = 4830;          % Front damping coefficient (Ns/m)
24 c2_ = 4830;          % Rear damping coefficient (Ns/m)
25 k1_ = 600*175.1*cosd(35);    % Front spring rate (N/m)
26 k2_ = 500*175.1*cosd(30);    % Rear spring rate (N/m)
27 m_ = 550;            % Mass of the vehicle (kg)
28 I_ = 550;            % Moment of interial about global x axis (kg*m^2)
29 a_ = 1.32;           % Distance from COG to front tire (m)
30 b_ = 1.28;           % Distance from COG to rear tire (m)
31
32 % ----- Forcing Function Parameters ----- %
33
34 amplitude = 0.015;      % Amplitude of bumps on road surface
35 bumpSparation = 2;       % Distance between peaks on the road surface (m)
36 speed = 22.2;           % Cruising speed (m/s)
37 wb = 2*pi*speed/bumpSparation; % Frequency of base excitation (rad/sec)
38 timeDelay = (a_+b_)/speed; % delay between input striking front and
39                           % rear wheels (s)
40
41 % Update simulink parameters
42 set_param('halfCarModel/DelayA','DelayTime',mat2str(timeDelay));
43 set_param('halfCarModel/DelayB','DelayTime',mat2str(timeDelay));
44
45 % ----- Create Symbolic Transfer Functions ----- %
46
47 syms t s c1 c2 k1 k2 m I a b t0 A w0 c k
48
49 QA1 = m*s^2 + (c1+c2)*s + (k1+k2); % X terms of ODE (A)
50 QA2 = (b*c2-a*c1)*s + (b*k2-a*k1); % Theta terms of ODE (A)
51 QB1 = (b*c2-a*c1)*s + (b*k2-a*k1); % X terms of ODE (B)
52 QB2 = I*s^2 + (b^2*c2+a^2*c1)*s + (b^2*k2+a^2*k1); % Theta terms of ODE (B)

```

```

53
54 PA1 = (c1*s+k1); % Non time shifted component of P for ODE (A) (front tire)
55 PA2 = (c2*s+k2); % Time shifted component of P for ODE (A) (rear tire)
56 PB1 = (-a*c1*s-a*k1); % Non time shifted component of P for ODE (B) (front tire)
57 PB2 = (b*c2*s+b*k2); % Time shifted component of P for ODE (B) (rear tire)
58
59 % Create linear systems in to solve for transfer functions
60
61 Q = [QA1, QA2;
62     QB1, QB2];
63
64 P1 = [PA1;
65     PB1];
66 P2 = [PA2;
67     PB2];
68
69 H1 = linsolve(Q,P1);
70 H2 = linsolve(Q,P2);
71
72 % Input in frequency domain
73 phaseShift = exp(-s*timeDelay); % Time shift to be applied to P_2
74 input = (amplitude*wb)/(s^2+wb^2); % Laplace transform of input L{A*sin(wb*t)}}
75
76 % ----- Solve for response -----
77
78 simplifiedH1 = simplify(collect(subs(H1, {k1 k2 c1 c2 m I a b}, {k1_ k2_ c1_ c2_
    m_ I_ a_ b_})));
79 simplifiedH2 = simplify(collect(subs(H2, {k1 k2 c1 c2 m I a b}, {k1_ k2_ c1_ c2_
    m_ I_ a_ b_})));
80 impulseResponse = vpa(simplify(ilaplace(simplifiedH1) + ilaplace(simplifiedH2*
    phaseShift)),3);
81
82 Fs = 200; % sampling frequency in Hz
83 T = 1/Fs; % sampling period
84 t_ = 0:T:30; % time values 0 to 30 seconds
85 L = length(t_);
86
87 % Numerical impulse response with zero damping
88 numericalImpulseResponse = double(subs(impulseResponse,t,t_));
89 figure;
90 hold on;
91 plot(t_,numericalImpulseResponse(1,:).*1000,'DisplayName','x_0');
92 ylabel('Linear Displacement (mm)');
93 yyaxis right
94 plot(t_,numericalImpulseResponse(2,:).*(180/pi),'DisplayName','\theta');
95 ylabel('Angular Displacement (deg)');
96 xlabel('Time (s)');
97 xlim([0 1]);
98 title('Impulse Response');
99 legend show;
100 saveas(gcf,'impResp','png');
101
102 % Compute the first harmonic of the displacement
103 fftOut = fft(numericalImpulseResponse(1,:));
104 p2 = abs(fftOut/L);
105 p1 = p2(1:L/2+1);

```

```

106 p1(2:end-1) = 2*p1(2:end-1);
107 f = Fs*(0:(L/2))/L;
108 figure;
109 hold on
110 plot(f,p1,'DisplayName','x_0');
111
112 % FourierTransform
113 fftOut = fft(numericalImpulseResponse(2,:));
114 p2 = abs(fftOut/L);
115 p1 = p2(1:L/2+1);
116 p1(2:end-1) = 2*p1(2:end-1);
117 f = Fs*(0:(L/2))/L;
118 plot(f,p1,'DisplayName','\theta');
119
120 legend show
121 xlabel('Frequency (Hz)');
122 title('FFT of Impusle Response');
123 xlim([0 10]);
124 saveas(gcf,'fftImpResp','png');
125
126 % ----- Forces -----
127
128 % ----- Simulink -----
129
130
131 numericalH1 = subs(H1, {c1 c2 k1 k2 m I a b}, {c1_ c2_ k1_ k2_ m_ I_ a_ b_});
132 numericalH2 = subs(H2, {c1 c2 k1 k2 m I a b}, {c1_ c2_ k1_ k2_ m_ I_ a_ b_});
133
134 [numH1, denomH1] = numden(numericalH1);
135 [numH2, denomH2] = numden(numericalH2);
136
137 transFuncA1 = tf(sym2poly(numH1(1)),sym2poly(denomH1(1)));
138 set_param('halfCarModel/HA1','Numerator',mat2str(sym2poly(numH1(1)))...
139 , 'Denominator',mat2str(sym2poly(denomH1(1))));
140
141 transFuncA2 = tf(sym2poly(numH2(1)),sym2poly(denomH2(1)));
142 set_param('halfCarModel/HA2','Numerator',mat2str(sym2poly(numH2(1)))...
143 , 'Denominator',mat2str(sym2poly(denomH2(1))));
144
145 transFuncB1 = tf(sym2poly(numH1(2)),sym2poly(denomH1(2)));
146 set_param('halfCarModel/HB1','Numerator',mat2str(sym2poly(numH1(2)))...
147 , 'Denominator',mat2str(sym2poly(denomH1(2))));
148
149 transFuncB2 = tf(sym2poly(numH2(2)),sym2poly(denomH2(2)));
150 set_param('halfCarModel/HB2','Numerator',mat2str(sym2poly(numH2(2)))...
151 , 'Denominator',mat2str(sym2poly(denomH2(2))));
152
153 % ----- Solving for natural resonance frequency -----

```

B MATLAB Source Code for Alternative Formulation

```

1 % ----- %
2 %
3 % Suspension Spring Rates Project - simplifiedFormulation.m
4 % Function to setup simplified transfer functions for a half car model to
5 % determine the response of MSXII to disturbances in the road conditions
6 %
7 % Author: Devon Copeland
8 %
9 % Midnight Sun 2017
10 %
11 %
12 % Notes:
13 %
14 % -
15 % ----- %
16
17 close all;
18 clear all;
19 clc;
20
21 % ----- Constants ----- %
22
23 c_ = 4830;          % Front damping coefficient (Ns/m)
24 k_ = 88600;         % Front spring rate (N/m)
25 M_ = 550;           % Mass of the vehicle (kg)
26 I_ = 550;           % Moment of interial about global x axis (kg*m^2)
27 a_ = 1.3;           % Distance from COG to front tire and rear tire (m)
28 zx_ = 0.49;          % Damping ratio of translational
29 wnx_ = 17.95;        % Natural frequency of pitch
30 zth_ = 0.64;          % Damping ratio of pitch
31 wnth_ = 23.33;        % Natural frequency of pitch
32
33
34 % ----- Forcing Function Parameters ----- %
35
36 A_ = 0.015;          % Amplitude of bumps on road surface
37 bumpSparation = 2;      % Distance between peaks on the road surface (m)
38 speed = 22.2;          % Cruising speed (m/s)
39 wb_ = 2*pi*speed/bumpSparation; % Frequency of base excitation (rad/sec)
40 t0_ = (2*a_)/speed;      % delay between input striking front and
                           % rear wheels (s)
41
42
43 % ----- Create Symbolic Transfer Functions ----- %
44
45 syms t s M I a t0 A wb c k zx wnx zth wnth
46
47 QA = s^2 + 2*c/M*s + wnx^2; % ODE A
48 QB = s^2 + 2*a^2*c/I*s + wnx^2; % ODE B
49
50 phaseShift = exp(-s*t0);
51
52 PA = [ (c*A*wb/M) * (s/(s^2+wb^2)) ;

```

```

53      (c*A*wb/M) * (s/(s^2+wb^2)) *phaseShift;
54      (k*A/M) * (wb/(s^2+wb^2));
55      (k*A/M) * (wb/(s^2+wb^2)) *phaseShift];
56
57 PB = [ (-c*a*A*wb/I)*(s/(s^2+wb^2));
58      (c*a*A*wb/I)*(s/(s^2+wb^2)) *phaseShift;
59      (-k*a*A/I)*(wb/(s^2+wb^2));
60      (k*a*A/I)*(wb/(s^2+wb^2)) *phaseShift];
61
62 X = collect((PA./QA),s);
63 Theta = collect((PB./QB),s);
64
65 subX = simplify(subs(X, {M I a t0 A wb c k zx wnx zth wnth}, {M_ I_ a_ t0_ A_ wb_
   c_ k_ zx_ wnx_ zth_ wnth_}));
66 subTheta = simplify(subs(Theta, {M I a t0 A wb c k zx wnx zth wnth}, {M_ I_ a_ t0_
   A_ wb_ c_ k_ zx_ wnx_ zth_ wnth_}));
67
68 x = simplify(collect(sum(ilaplace(subX))));
69 theta = simplify(collect(sum(ilaplace(subTheta))));
70
71 % Print total response
72 latex(vpa(x,3))
73 latex(vpa(theta,3))
74
75 totalResponse = [x;theta];
76
77 % Plot total response
78 t_ = 0:0.005:1;
79 numericalResponse = double(subs(totalResponse,t,t_));
80 figure;
81 hold on;
82 plot(t_,numericalResponse(1,:).*1000,'DisplayName','x_0');
83 ylabel('Linear Displacement (mm)');
84 yyaxis right
85 plot(t_,numericalResponse(2,:).*(180/pi),'DisplayName','\theta');
86 ylabel('Angular Displacement (deg)');
87 xlabel('Time (s)');
88 xlim([0 1]);
89 title('Total Response');
90 legend show;
91 saveas(gcf,'totalResp','png');

```