Midnight Sun Solar Car Team
University of Waterloo

# MSXII Suspension Parameters

Prepared for Project 3 of ME321 Winter 2017

Prepared by:
Devon Copeland

June 4, 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 | $m$ |
| Distance from Front Wheel to CoG ($a$) | 1.32 | $m$ |
| Distance from Road to CoG ($h$) | 0.52 | $m$ |
| Moment of Inertia Normal to Symmetry Plane at CoG ($I_{xx}$) | 550 | $kgm^2$ |
| 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.

(a) Front Isometric View (b) Rear Isometric View (c) Rear Sie View

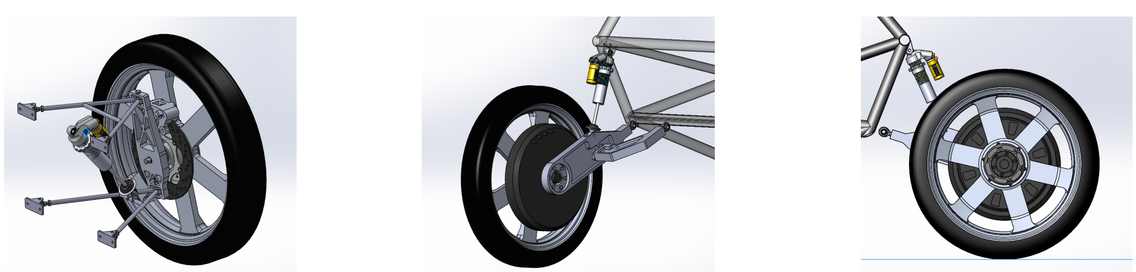

Figure 1: MSXII Suspension Design (Screenshots from Solidworks)

## 1.4 Selecting Suspension Parameters

The spring rates are calculated based on the travel of the wheel and the maximum vertical loading of the wheel due to normal force from the road. Assuming a smooth driving surface, the largest normal force acting on the tire will occur while braking in a turn. To approximate this loading condition, the a superposition of two static half car models is used; one for cornering and one for hard braking.

### 1.4.1 Hard Braking

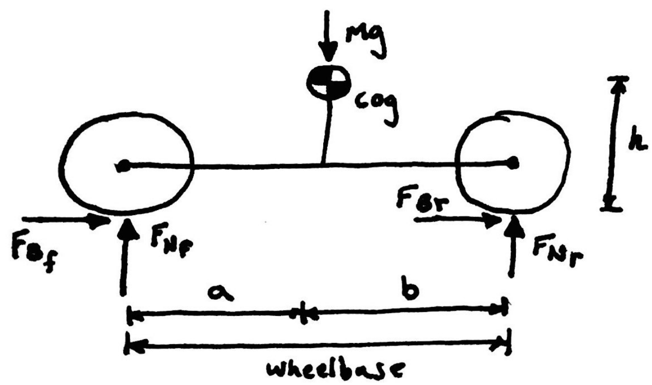


Figure 2: Free Body Diagram of a Car Braking

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, $\mu_s$, 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:

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

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

$$aF_{Nf} = hF_{braking} + bF_{Nr} \tag{2}$$

$$F_{Nf} + F_{Nr} = Mg \tag{3}$$

Solving for the front normal force:

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

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.

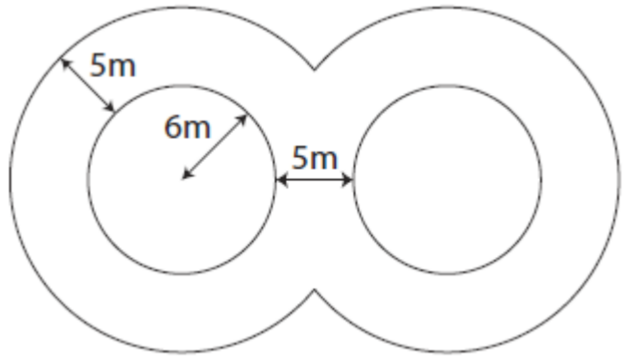


Figure 3: Figure of Eight Course from ASC 2018 Regulations [1]

### 1.4.2 Cornering

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 cornering force required to navigate the figure of eight can be found as follows:

$$F_{si} + F_{so} = F_{steer} = \frac{Mv^2}{r} = \frac{(550kg)\left(\frac{34\pi m}{18s}\right)^2}{8.5m} = 2.28kN \tag{5}$$

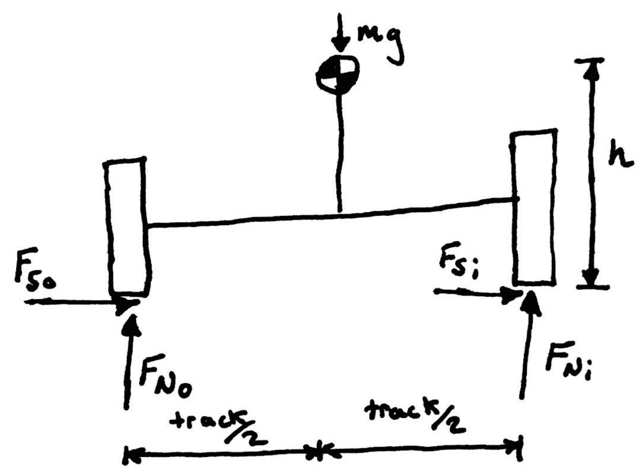


Figure 4: Free Body Diagram of a Car Cornering

For the half car model shown if Figure 4, summing the moments about the centre of gravity and the vertical forces results in the following equations:

$$\frac{track}{2}F_{No} = hF_{steer} + \frac{track}{2}F_{Ni} \tag{6}$$

$$F_{No} + F_{Ni} = Mg \tag{7}$$

Solving for the outside wheel normal force:

$$
\begin{aligned}
\frac{track}{2} F_{No} &= hF_{steer} + \frac{track}{2}(Mg - F_{Ni}) \\
F_{No} &= \frac{hF_{steer} + \frac{track}{2} Mg}{track} = \frac{(0.52m)(2.28kN) + (0.80m)(550kg)\left(9.81\frac{m}{s^2}\right)}{1.60m} = 3.44kN
\end{aligned}
\tag{8}
$$

Note that the 3.44kN above is for both front and rear outside wheels. Assuming the force is evenly split between the front and rear, there is approximately **1.72kN** of normal force on each outside tire during maximum cornering.

### 1.4.3 Superposition

Both the braking and cornering conditions subject the front outside wheel to a normal force greater than the nominal normal force seen when the car is at rest. The change in normal force for each loading condition can be expressed as follows:

$$
\begin{aligned}
\Delta F_{N_{braking}} &= F_{Nf_{braking}} - \frac{Mg}{4} = 1.87kN - 1.35kN = 0.52kN \\
\Delta F_{N_{cornering}} &= F_{No_{cornering}} - \frac{Mg}{4} = 1.72kN - 1.35kN = 0.37kN
\end{aligned}
\tag{9}
$$

To approximate the worst case loading condition, a superposition of the changes in normal force and the nominal normal force is applied.

$$
F_{N_{max}} = \Delta F_{N_{braking}} + \Delta F_{N_{cornering}} + F_{N_{nominal}} = 0.52kN + 0.37kN + 1.35kN = 2.24kN \tag{10}
$$

### 1.4.4 Spring Rate Selection

The maximum travel of the suspension for MSXII in compression has been set to 45mm. It is desired that under the combined loading condition of braking and cornering, the suspension compresses to a maximum of 55% of the travel. As such, the desired spring rate is found to be:

$$
k = \frac{force}{displacement} = \frac{2.24kN}{(55\%)(0.045m)} = 88.6kN/m \tag{11}
$$

Note that the above rate assumes a vertical spring. The vertical spring rate can be mapped to an equivalent angled spring rate by dividing by the cosine of the angle that the coilover makes with the vertical direction:

$$
\begin{aligned}
k_{front} &= \frac{88.6}{cos(35)} = 108.2kN/m \\
k_{rear} &= \frac{88.6}{cos(30)} = 102.3kN/m
\end{aligned}
\tag{12}
$$

MSXII will be using Ohlins TTX25 MkII coilovers for both the front and rear. These coilovers can be configured to have spring rates from 150 lb/in to 650 lb/in in increments of 50 lb/in. 1 lb/in is approximately 175 N/m therefore 600 lb/in coils should be selected for the front and 600 lb/in coils for the rear.

### 1.4.5 Damping Coefficient Selection

Ohlins has published damper curves for their TTX25 MkII's however real world damping rates are non constant. For this report, the damping coefficient will be idealized as constant and will be selected such that the damping ratio, $\zeta$, is 0.5. This is typical of high performance consumer vehicles [2]. Given an equivalent vertical spring rate or 88.6/m, a damping ratio of 0.5 maps to a vertical damping coefficient of (4.83kNs/m).

# 2 Suspension Model

## 2.1 Overview

MSXII's suspension is modelled using a point mass connected to two vertical, linear spring mass damper systems at the two wheels. This half car model assumes infinite tire stiffness and that the vehicle pitches about it's center of gravity. Figure 5 describes the proposed model where:

$\theta$ is the pitch angle about the axis normal to the vehicle's symmetry plane

$x_0$ is the displacement of the center of gravity in the vertical axis

$x_1$ & $x_2$ are the displacements of the mounting points for the front and rear wheels respectively

$y_1$ & $y_2$ is the change in road elevation below the front and rear wheels respectively

$M$ is the mass of the vehicle

$I$ is the moment of inertia normal to the vehicles symmetry plane at the center of gravity

$k1, k2, c1$ & $c_2$ are the spring and damping coefficients

$a$ & $b$ are the distances from center of gravity to the front and rear wheels respectively

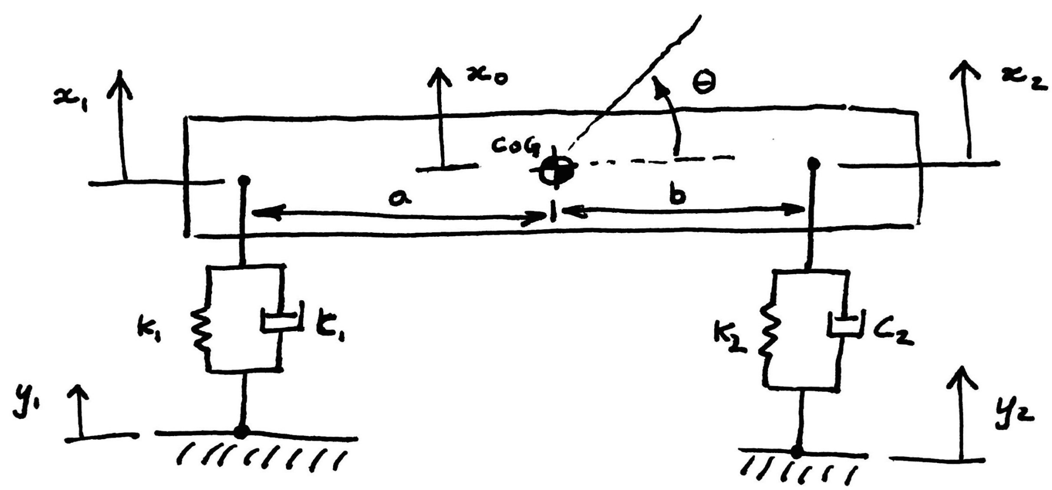


Figure 5: Half car model with infinite tire stiffness

## 2.2 Equations of Motion

### 2.2.1 Dynamic Equilibrium

The spring mass damper system shown in Figure 5 can be described by the folowing two differential equations:

$$M\ddot{x} + c_1(\dot{x}_1 - \dot{y}_1) + c_2(\dot{x}_2 - \dot{y}_2) + k_1(x_1 - y_1) + k_2(x_2 - y_2) = 0 \tag{13}$$

$$I\ddot{\theta} - ac_1(\dot{x}_1 - \dot{y}_1) + bc_2(\dot{x}_2 - \dot{y}_2) - ak_1(x_1 - y_1) + bk_2(x_2 - y_2) = 0 \tag{14}$$

Rearranging to move the inputs $y_1$ and $y_2$ to right hand side:

$$M\ddot{x} + c_1\dot{x}_1 + c_2\dot{x}_2 + k_1x_1 + k_2x_2 = c_1\dot{y}_1 + c_2\dot{y}_2 + k_1y_1 + k_2y_2 \tag{15}$$

$$I\ddot{\theta} - ac_1\dot{x}_1 + bc_2\dot{x}_2 - ak_1x_1 + bk_2x_2 = -ac_1\dot{y}_1 + bc_2\dot{y}_2 - ak_1y_1 + bk_2y_2 \tag{16}$$

By assuming small pitch angles, $\theta$, the angle between the spring mass damper system becomes approximately perpendicular and the displacements $x_1$ and $x_2$ can be approximated as follows:

$$x_1 = x_0 - a\theta \tag{17}$$

$$x_2 = x_0 + b\theta \tag{18}$$

It follows that the time derivatives of the above two equations are:

$$\dot{x}_1 = x_0 - a\dot{\theta} \tag{19}$$

$$\dot{x}_2 = x_0 + b\dot{\theta} \tag{20}$$

Substituting into the original differential equations:

$$M\ddot{x} + \dot{x}_0(c_2 + c_1) + \dot{\theta}(bc_2 - ac_1) + x_0(k_2 + k_1) + \theta(bk_2 - ak_1) = c_1\dot{y}_1 + c_2\dot{y}_2 + k_1y_1 + k_2y_2 \tag{21}$$

$$I\ddot{\theta} + \dot{x}_0(bc_2 - ac_1) + \dot{\theta}(b^2c_2 + a^2c_1) + x_0(bk_2 - ak_1) + \theta(b^2k_2 + a^2k_1) = -ac_1\dot{y}_1 + bc_2\dot{y}_2 - ak_1y_1 + bk_2y_2 \tag{22}$$

The above two equations are considerably complex and are not particularly conducive of solving analytically by hand. However, the fact that the vehicle has the center of gravity approximately centered along the wheel such that $a$ and $b$ are equal, offers an avenue for simplification. Additionally if the front and rear spring and damping coefficients are assumed to be the same ($k$ and $c$ respectively), the problem's complexity is drastically reduced as shown in the following two equations:

$$M\ddot{x} + 2c\dot{x}_0 + 2kx_0 = c\dot{y}_1 + c\dot{y}_2 + ky_1 + ky_2 \tag{23}$$

$$I\ddot{\theta} + 2a^2c\dot{\theta} + 2a^2k\theta = -ac\dot{y}_1 + bc_2\dot{y}_2 - ak_1y_1 + bk_2y_2 \tag{24}$$

### 2.2.2 Natural Frequency

The natural frequency of the translational and rotational components of the system are found as follows:

$$\begin{aligned} \omega_{nx} &= \sqrt{\frac{k_{eq}}{m_{eq}}} = \sqrt{\frac{2k}{M}} \\ \omega_{n\Theta} &= \sqrt{\frac{k_{eq}}{m_{eq}}} = \sqrt{\frac{2a^2k}{I}} \end{aligned} \tag{25}$$

### 2.2.3 Critical Damping

The critical camping coefficient for the translational and rotational components of the system are found as follows:

$$\begin{aligned} c_{eqx} = 2c, c_{rx} = 2m_{eq}\omega_n = \sqrt{8kM}, \zeta_x = \frac{c_{eqx}}{c_r} \\ c_{eq\Theta} = 2a^2c, c_{r\Theta} = 2m_{eq}\omega_n = \sqrt{8a^2kI}, \zeta_\Theta = \frac{c_{eq\Theta}}{c_r} \end{aligned} \tag{26}$$

### 2.2.4 Final Equations of Motion

The road displacement can be approximated by a sinusoid with and amplitude $A$ and a base excitation frequency $\omega_b$. Since the rear wheels experience the same displacements as the front $y_2$ can be expressed as a phase shifted version of $y_1$. Combining this with the natural frequency and damping ratio results in the following equations of motion:

$$\ddot{x}+(2\zeta_x M\omega_{nx})\dot{x}_0+(\omega_{nx}^2)x_0 = \frac{cA\omega_b}{M}cos(\omega_b t)+\frac{cA\omega_b}{M}cos(\omega_b(t-t_0))+\frac{kA}{M}sin(\omega_b t)+\frac{kA}{M}sin(\omega_b(t-t_0)) \tag{27}$$

$$\begin{aligned} &\ddot{\theta}+(2\zeta_\Theta I\omega_{n\Theta})\dot{\theta} + (\omega_{n\Theta}^2)\theta \\ &= -\frac{acA\omega_b}{I}cos(\omega_b t) + \frac{acA\omega_b}{I}cos(\omega_b(t-t_0)) - \frac{kcA\omega_b}{I}sin(\omega_b t) + \frac{akA\omega_b}{I}sin(\omega_b(t-t_0)) \end{aligned} \tag{28}$$

## 2.3 Total Response

### 2.3.1 Base Excitation Parameters

To model real road conditions, a sinusoid with a amplitude of 0.015 and a frequency of 69.7 rad/s is used. This corresponds to 30mm peak to peak bumps that are separated by a distance of 2m while driving at 80km/hr (calculated in Appendix A).

### 2.3.2 Initial conditions

For the total response, the initial conditions of the system are all set to zero. This represents the vehicle traveling on a perfectly smooth road before encountering disturbances at time $t = 0$.

### 2.3.3 Solution

Based on the idealize parameters proposed in Section 1.4 , the natural frequency and damping ratio are found as follows:

$$\begin{aligned} \omega_{nx} = \sqrt{\frac{2k}{M}} = \sqrt{\frac{(2)(88600N/m)}{550kg}} = 17.95rad/s = 2.86Hz \\ \omega_{n\Theta} = \sqrt{\frac{2a^2k}{I}} = \sqrt{\frac{(2)(1.3^2m^2)(88600N)}{550kgm^2}} = 23.33rad/s = 3.71Hz \end{aligned} \tag{29}$$

$$\begin{aligned} c_{eqx} &= 2c = (2)(4830Ns/m) = 9660Ns/m \\ c_{rx} &= 2M\omega_n = (2)(550kg)(17.75rad/s) = 19530Ns/m \\ \zeta_x &= \frac{c_{eqx}}{c_r} = \frac{9660Ns/m}{19530Ns/m} = 0.49 \\ c_{eq\Theta} &= 2a^2c = 2(1.3^2m^2)(4830Ns/m) = 16330Nms \\ c_{r\Theta} &= 2m_{eq}\omega_n = (2)(550kgm^2)(23.33rad/s) = 25660Nms \\ \zeta_\Theta &= \frac{c_{eq\Theta}}{c_r} = \frac{16330Nms}{25660Nms} = 0.64 \end{aligned} \tag{30}$$

Using the MATLAB script shown in Appendix A, values for the parameters $I$, $M$, $k$, $c$, and $a$ are substituted into Equations 27 and 28 and the Laplace transform is taken. Solving for $X(s)$ and $\Theta(s)$ and taking the inverse Laplace transform yeilds:

$$\begin{aligned} x(t) =& (3.12\cdot 10^{-5})\sin(69.7\,t) - 0.00201\cos(69.7\,t) + 1.0\,\mathrm{u}(1.0\,t - 0.117)\,((5.91\cdot 10^{-4})\cos(69.7\,t) \\ &- 0.00192\sin(69.7\,t)) + \mathrm{e}^{-8.78\,t}\,(0.00201\cos(15.2\,t) + 0.00102\sin(15.2\,t) \\ &- \mathrm{u}(1.0\,t - 0.117)\,(0.00396\cos(15.2\,t) - 0.00489\sin(15.2\,t))) \end{aligned} \tag{31}$$

$$\begin{aligned} \theta(t) =& 0.00242\cos(69.7\,t) + (3.31\cdot 10^{-4})\,\mathrm{u}(1.0\,t - 0.117)\cos(69.7\,t) - 0.00242\,\mathrm{e}^{-14.8\,t}\cos(9.42\,t) \\ &- (5.64\cdot 10^{-4})\,\mathrm{e}^{-14.8\,t}\sin(9.42\,t) - 1.0\sin(69.7\,t)\,(0.00244\,\mathrm{u}(1.0\,t - 0.117) + 4.39\cdot 10^{-4}) \\ &+ 0.00334\,\mathrm{u}(1.0\,t - 0.117)\,\mathrm{e}^{-14.8\,t}\cos(9.42\,t) + 0.0137\,\mathrm{u}(1.0\,t - 0.117)\,\mathrm{e}^{-14.8\,t}\sin(9.42\,t) \end{aligned} \tag{32}$$

### 2.3.4 Plotting Total Response

Figure 6 shows the total response of the system subjected to the simulated road input.

## 2.4 Alternate Formulation

If the center of gravity is not centered between the front and rear wheel axes, the problem increases significantly in complexity and an alternate approach to solving the problem is required. By taking the unilateral Laplace transform of the above two equations with all initial conditions set to zero, the following $s$ domain equations are obtained:

$$\left[Ms^2 + (c_2 + c_1)s + (k_2 + k_1)\right]X_0 + \left[(bc_2 - ac_1)s + (bk_2 + ak_1)\right]\Theta = c_1sY_1 + c_2sY_2 + k_1Y_1 + k_2Y_2 \tag{33}$$

$$\left[(bc_2 - ac_1)s + (bk_2 - ak_1)\right]X_0 + \left[Is^2 + (b^2c_2 + a^2c_1)s + (b^2k_2 + a^2k_1)\right]\Theta = c_1sY_1 + c_2sY_2 + k_1Y_1 + k_2Y_2 \tag{34}$$

The above two equations can be expressed in matrix form as follows:

$$\begin{bmatrix} Q_{Ax}(s) & Q_{A\Theta}(s) \\ Q_{Bx}(s) & Q_{B\Theta}(s) \end{bmatrix} \begin{bmatrix} X_0 \\ \Theta \end{bmatrix} = \begin{bmatrix} P_{A1}(s) \\ P_{B1}(s) \end{bmatrix} Y_1 + \begin{bmatrix} P_{A2}(s) \\ P_{B2}(s) \end{bmatrix} Y_2 \tag{35}$$

If it is assumed that the rear tire of the car experiences the exact same road conditions as the front tire, $Y1$ and $Y2$ can be expressed as functions of the same input, $Y$ where $Y2$ is simply a phase shifted version of $Y1$:

$$\begin{aligned} y_1(t) &= y(t) \\ y_2(t) &= y(t - t_0) \end{aligned} \implies \begin{aligned} Y_1(s) &= Y(s) \\ Y_2(s) &= Y(s)e^{-st_0} \end{aligned} \tag{36}$$

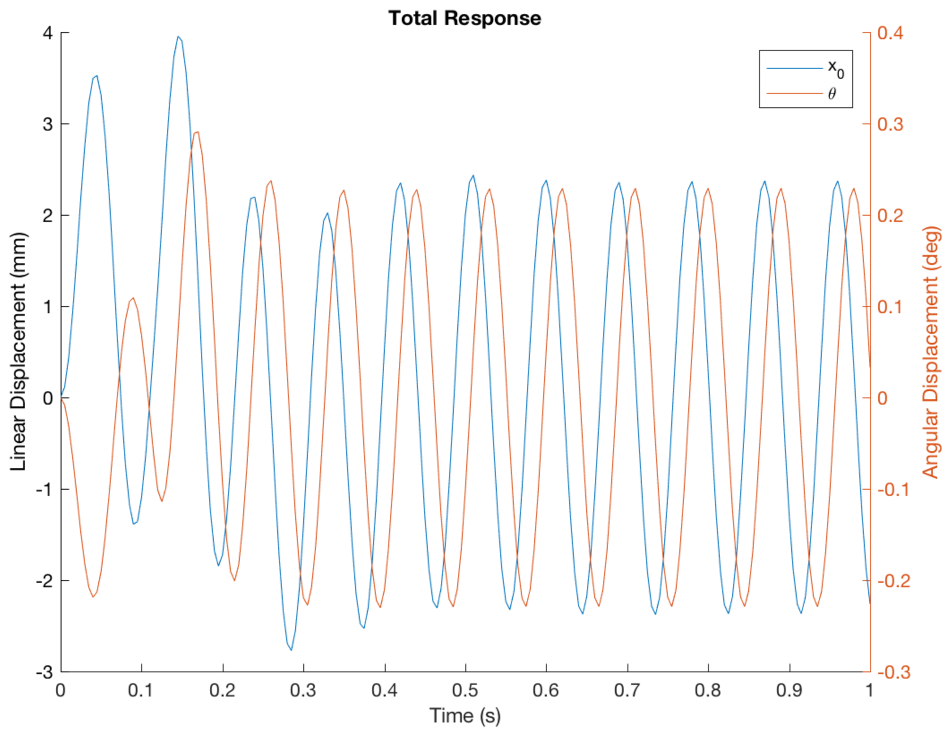


Figure 6: Total Response of the System

Equation 37 can then be rewritten as follows:

$$Q\begin{bmatrix} X_0 \\ \Theta \end{bmatrix} = \left([P_1] + [P_2]\, e^{-st_0}\right) Y \tag{37}$$

Since the system is linear, the above matrix equation can be solved in two parts to obtain the transfer functions for $X_0$ and $\Theta$:

$$\frac{\begin{bmatrix} X_0 \\ \Theta \end{bmatrix}}{Y} = [H] = [H_1] + [H_2] = [Q^{-1}][P_1] + [Q^{-1}][P_1]e^{-st_0} \tag{38}$$

Using the MATLAB script shown in Appendix B, the transfer function $H(s)$ is solved for symbolically as described above and the parameters for $I$, $M$, $k_1$, $k_2$, $c_1$, $c_2$, $a$ and $b$ are substituted into the equation. The inverse laplace transform is then taken to obtain the impulse response as shown in Figure 7.

By performing an Fast Fourier Transform on the undamped system the harmonics of the system can be approximated as shown in Figure 8. A sampling frequency of 200Hz and and a window of 30s was used for the FFT. From Figure 8, the largest harmonics of the system occur at approximately 2Hz for translational displacement and 4Hz for angular displacement. Note that these frequencies align closely with the natural frequencies that were determined analytically in the simplified case.

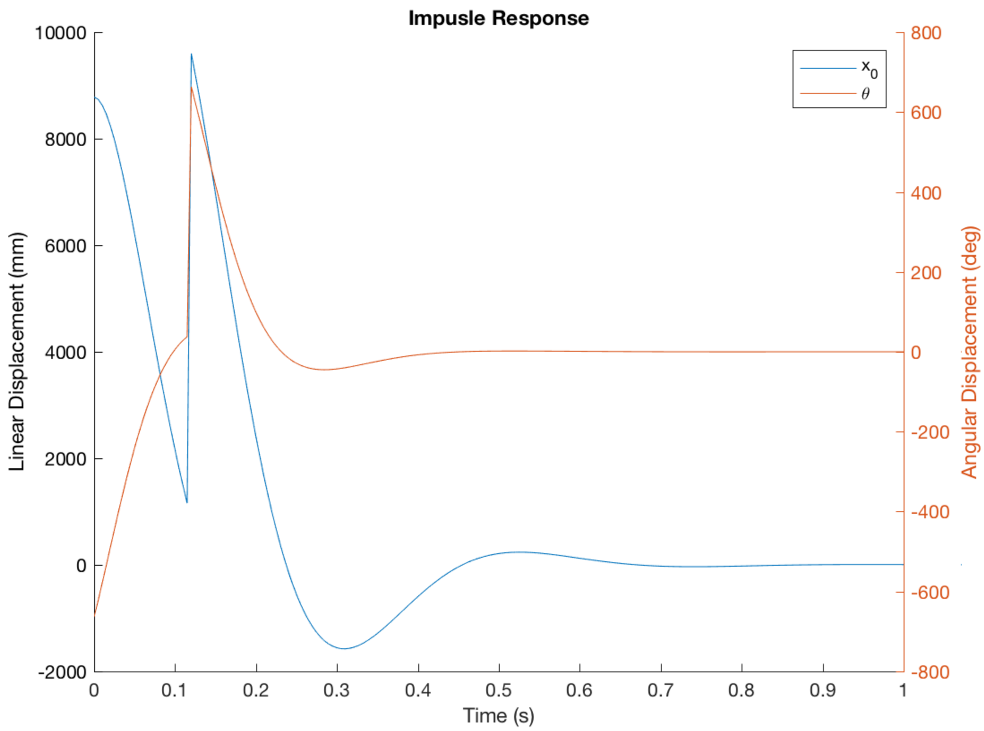


Figure 7: Undamped Impulse Response

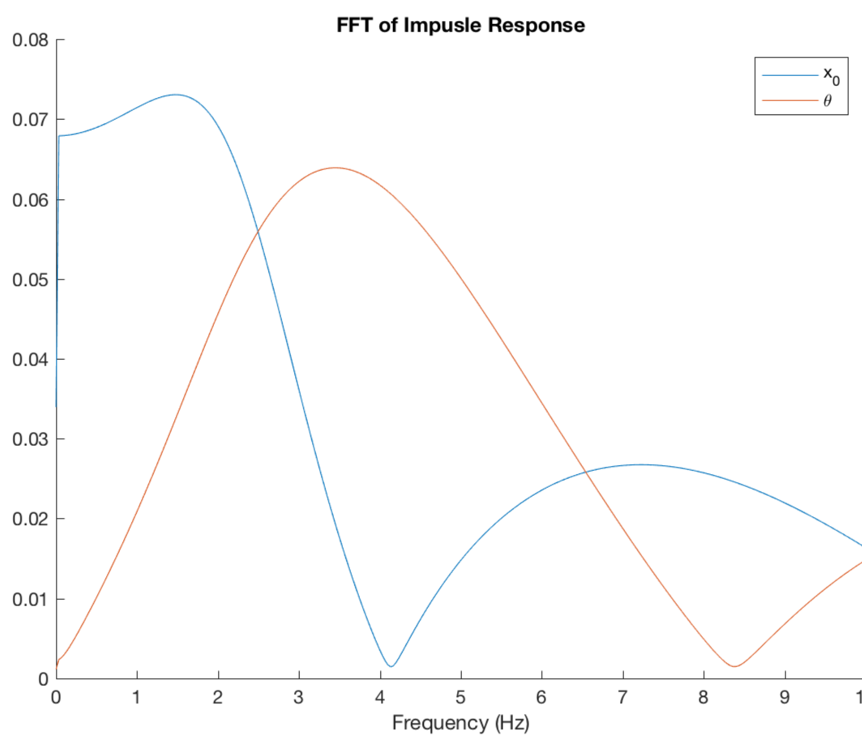


Figure 8: FFT of the Undamped Impulse Response

# 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] Jim Kasprzak. Understanding your dampers: A guide from jim kasprzak, [online], 2017. http://www.kaztechnologies.com/wp-content/uploads/2014/03/A-Guide-To-Your-Dampers-Chapter-from-FSAE-Book-by-Jim-Kasprzak.pdf.

# A MATLAB Source Code for Simplified Model

```
% ------------------------------------------------------------------------ %
%
% Suspension Spring Rates Project - alternativeFormulation.m
% Function to setup transfer functions for a half car model to determine the
% reponse of MSXII to disturbances in the road conditions
%
% Author: Devon Copeland
%
%   Midnight Sun 2017
%
%
% Notes:
%
%   -
% ------------------------------------------------------------------------ %

load_system('halfCarModel');
close all;
clear all

% ------------------------------ Constants ------------------------------- %

c1_ = 4830;      % Front damping coefficient (Ns/m)
c2_ = 4830;      % Rear damping coefficient (Ns/m)
k1_ = 600*175.1*cosd(35);     % Front sring rate (N/m)
k2_ = 500*175.1*cosd(30);     % Rear sring rate (N/m)
m_ = 550;        % Mass of the vehicle (kg)
I_ = 550;        % Moment of interial about global x axis (kg*m^2)
a_ = 1.32;        % Distance from COG to front tire (m)
b_ = 1.28;        % Distance from COG to rear tire (m)

% ---------------------- Forcing Function Parameters -------------------- %

amplitude = 0.015;                % Amplitude of bumps on road surface
bumpSparation = 2;                % Distance between peaks on the road surface (m)
speed = 22.2;                     % Cruising speed (m/s)
wb = 2*pi*speed/bumpSparation;    % Frequency of base excitation (rad/sec)
timeDelay = (a_+b_)/speed;        % delay between input striking front and
                                  % rear wheels (s)

% Update simulink parameters
set_param('halfCarModel/DelayA','DelayTime',mat2str(timeDelay));
set_param('halfCarModel/DelayB','DelayTime',mat2str(timeDelay));

% ----------------- Create Symbolic Transfer Functions ----------------- %

syms t s c1 c2 k1 k2 m I a b t0 A w0 c k

QA1 = m*s^2 + (c1+c2)*s + (k1+k2); % X terms of ODE (A)
QA2 = (b*c2-a*c1)*s + (b*k2-a*k1); % Theta terms of ODE (A)
QB1 = (b*c2-a*c1)*s + (b*k2-a*k1); % X terms of ODE (B)
QB2 = I*s^2 + (b^2*c2+a^2*c1)*s + (b^2*k2+a^2*k1); % Theta terms of ODE (B)
```

```
PA1 = (c1*s+k1); % Non time shifted componet of P for ODE (A) (front tire)
PA2 = (c2*s+k2); % Time shifted componet of P for ODE (A) (rear tire)
PB1 = (-a*c1*s-a*k1); % Non time shifted componet of P for ODE (B) (front tire)
PB2 = (b*c2*s+b*k2); % % Time shifted componet of P for ODE (B) (rear tire)

% Create linear systems in to solve for transfer functions

Q = [QA1, QA2;
     QB1, QB2];

P1 = [PA1;
      PB1];
P2 = [PA2;
      PB2];

H1 = linsolve(Q,P1);
H2 = linsolve(Q,P2);

% Input in frequency domain
phaseShift = exp(-s*timeDelay); % Time shift to be applied to P_2
input = (amplitude*wb)/(s^2+wb^2); % Laplace transform of input L{A*sin(wb*t))}

% -------------------------- Solve for response  ----------------------- %

simplifiedH1 = simplify(collect(subs(H1, {k1 k2 c1 c2 m I a b}, {k1_ k2_ c1_ c2_ m_ I_ a_ b_})));
simplifiedH2 = simplify(collect(subs(H2, {k1 k2 c1 c2 m I a b}, {k1_ k2_ c1_ c2_ m_ I_ a_ b_})));
impulseResponse = vpa(simplify(ilaplace(simplifiedH1) + ilaplace(simplifiedH2*phaseShift)),3);

Fs = 200; % sampling frequency in Hz
T = 1/Fs; % sampling period
t_ = 0:T:30; % time values 0 to 30 seconds
L = length(t_);

% Numerical impulse response with zero damping
numericalImpulseResponse = double(subs(impulseResponse,t,t_));
figure;
hold on;
plot(t_,numericalImpulseResponse(1,:).*1000,'DisplayName','x_0');
ylabel('Linear Displacement (mm)');
yyaxis right
plot(t_,numericalImpulseResponse(2,:).*(180/pi),'DisplayName','\theta');
ylabel('Angular Displacement (deg)');
xlabel('Time (s)');
xlim([0 1]);
title('Impusle Response');
legend show;
saveas(gcf,'impResp','png');

% Compute the first harmonic of the displacement
fftOut = fft(numericalImpulseResponse(1,:));
p2 = abs(fftOut/L);
p1 = p2(1:L/2+1);
```

```
p1(2:end-1) = 2*p1(2:end-1);
f = Fs*(0:(L/2))/L;
figure;
hold on
plot(f,p1,'DisplayName','x_0');

% FourierTransform
fftOut = fft(numericalImpulseResponse(2,:));
p2 = abs(fftOut/L);
p1 = p2(1:L/2+1);
p1(2:end-1) = 2*p1(2:end-1);
f = Fs*(0:(L/2))/L;
plot(f,p1,'DisplayName','\theta');

legend show
xlabel('Frequency (Hz)');
title('FFT of Impusle Response');
xlim([0 10]);
saveas(gcf,'fftImpResp','png');

% --------------------------------- Forces ----------------------------- %


% ------------------------------- Simulink ----------------------------- %

numericalH1 = subs(H1, {c1 c2 k1 k2 m I a b}, {c1_ c2_ k1_ k2_ m_ I_ a_ b_});
numericalH2 = subs(H2, {c1 c2 k1 k2 m I a b}, {c1_ c2_ k1_ k2_ m_ I_ a_ b_});

[numH1, denomH1] = numden(numericalH1);
[numH2, denomH2] = numden(numericalH2);

transFuncA1 = tf(sym2poly(numH1(1)),sym2poly(denomH1(1)));
set_param('halfCarModel/HA1','Numerator',mat2str(sym2poly(numH1(1)))...
                            ,'Denominator',mat2str(sym2poly(denomH1(1))));

transFuncA2 = tf(sym2poly(numH2(1)),sym2poly(denomH2(1)));
set_param('halfCarModel/HA2','Numerator',mat2str(sym2poly(numH2(1)))...
                            ,'Denominator',mat2str(sym2poly(denomH2(1))));

transFuncB1 = tf(sym2poly(numH1(2)),sym2poly(denomH1(2)));
set_param('halfCarModel/HB1','Numerator',mat2str(sym2poly(numH1(2)))...
                            ,'Denominator',mat2str(sym2poly(denomH1(2))));

transFuncB2 = tf(sym2poly(numH2(2)),sym2poly(denomH2(2)));
set_param('halfCarModel/HB2','Numerator',mat2str(sym2poly(numH2(2)))...
                            ,'Denominator',mat2str(sym2poly(denomH2(2))));

% --------------- Solving for natural resonance frequency -------------- %
```

# B MATLAB Source Code for Alternative Formulation

```
% ----------------------------------------------------------------------- %
%
% Suspension Spring Rates Project - simplifiedFormulation.m
% Function to setup simplified transfer functions for a half car model to
% determine the reponse of MSXII to disturbances in the road conditions
%
% Author: Devon Copeland
%
%   Midnight Sun 2017
%
%
% Notes:
%
%   -
% ----------------------------------------------------------------------- %

close all;
clear all;
clc;

% ----------------------------- Constants ------------------------------ %

c_ = 4830;      % Front damping coefficient (Ns/m)
k_ = 88600;     % Front sring rate (N/m)
M_ = 550;       % Mass of the vehicle (kg)
I_ = 550;       % Moment of interial about global x axis (kg*m^2)
a_ = 1.3;       % Distance from COG to front tire and rear tire (m)
zx_ = 0.49;      % Damping ratio of translational
wnx_ = 17.95;    % Natural frequency of pitch
zth_ = 0.64;     % Damping ratio of pitch
wnth_ = 23.33;   % Natural frequency of pitch


% ---------------------- Forcing Function Parameters ------------------- %

A_ = 0.015;                       % Amplitude of bumps on road surface
bumpSparation = 2;                % Distance between peaks on the road surface (m)
speed = 22.2;                     % Cruising speed (m/s)
wb_ = 2*pi*speed/bumpSparation;   % Frequency of base excitation (rad/sec)
t0_ = (2*a_)/speed;              % delay between input striking front and
                                  % rear wheels (s)

% ----------------- Create Symbolic Transfer Functions ----------------- %

syms t s M I a t0 A wb c k zx wnx zth wnth

QA = s^2 + 2*c/M*s + wnx^2; % ODE A
QB = s^2 + 2*a^2*c/I*s + wnx^2; % ODE B

phaseShift = exp(-s*t0);

PA = [(c*A*wb/M)*(s/(s^2+wb^2));
```

```
       (c*A*wb/M)*(s/(s^2+wb^2))*phaseShift;
       (k*A/M)*(wb/(s^2+wb^2));
       (k*A/M)*(wb/(s^2+wb^2))*phaseShift];

PB = [(-c*a*A*wb/I)*(s/(s^2+wb^2));
       (c*a*A*wb/I)*(s/(s^2+wb^2))*phaseShift;
       (-k*a*A/I)*(wb/(s^2+wb^2));
       (k*a*A/I)*(wb/(s^2+wb^2))*phaseShift];

X = collect((PA./QA),s);
Theta = collect((PB./QB),s);

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_}));
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_}));

x = simplify(collect(sum(ilaplace(subX))));
theta = simplify(collect(sum(ilaplace(subTheta))));

% Print total response
latex(vpa(x,3))
latex(vpa(theta,3))

totalResponse = [x;theta];

% Plot total response
t_ = 0:0.005:1;
numericalResponse = double(subs(totalResponse,t,t_));
figure;
hold on;
plot(t_,numericalResponse(1,:).*1000,'DisplayName','x_0');
ylabel('Linear Displacement (mm)');
yyaxis right
plot(t_,numericalResponse(2,:).*(180/pi),'DisplayName','\theta');
ylabel('Angular Displacement (deg)');
xlabel('Time (s)');
xlim([0 1]);
title('Total Response');
legend show;
saveas(gcf,'totalResp','png');
```