MECH 888: Nonlinear Optimization
Professor: Dr. Florin Bobaru
Author: Ricardo Jacome

Road Curvature Model Optimization

Abstract

With the advancement of autonomous vehicle technology, the use of sensors has become the cornerstone of any autonomous navigation architecture. However, limitations such as weather disruptions, often make sensors to be unreliable and even have severe consequences. A new method is proposed in which a guidance data profile is generated independently of vehicle sensors to provide vehicles with enough information to traverse curves. This method relies on vehicle dynamics and street design standards. An optimization routine is implemented to obtain optimized guidance parameters such as curvature, velocity, or wheel angle. This optimization routine is divided into two main optimization problems. The results show that the guidance profiles generated can be optimized with reasonable values for providing a ride that can complement existing autonomous vehicle technology.

Background

In the United States, the prevailing standards for road design come from The American Association of State Highway and Transportation Officials, referred as the Green Book. This book offers an extensive review of road design considerations that comply with vehicle dynamic behavior. Special consideration is given to curve maneuvering because there are centripetal forces that need to be balanced with a combination of road factors to maintain vehicle stability. The dynamics are formulated using Newton's Second Law of motion, which takes into consideration both road and vehicle characteristics. Road design parameters include, road friction, superelevation, and maximum width, while the vehicle parameters considered are velocity, acceleration, trackwidth, and vehicle length. These are summarized with the following formula [1]:

$$\frac{v^2}{g\rho} = \frac{\mu + 0.01e}{1 - 0.01\mu e} \qquad Eq. 1$$

Where:

$v =$ Vehicle velocity (m/s)	$\mu =$Coefficient of side road friction
$e =$Superelevation (as a percentage)	$\rho =$ Radius of curvature (m)
$g =$ Gravitational acceleration (9.81 m/s^2)	

This formula relates most of the parameters that can be involved from any general vehicle and general street. In the case of street design, road friction and superelevation are already implemented on most roads. Vehicles can vary their velocity and heading accordingly as they traverse any curve. However, the factor that unites both the road and the vehicle is the radius of curvature. From geometric considerations and Newton's Second Law, it is possible to find another equation that relates more vehicle parameters to the radius of curvature as follows [2]:

$$\delta = (57.3L + \overline{UG}v^2)\rho^{-1} \qquad Eq. 2$$

Where:

$\delta =$ Wheel directional angle (deg)	L = Vehicle length (m)
$\rho =$ Radius of curvature (m)	$\overline{UG} =$ Understeer gradient (deg-s^2/m)
$v =$ Vehicle velocity (m/s)	

Since the Radius of Curvature belongs to the set of all real numbers, to avoid division by zero, the inverse function is used. This is known as Curvature and is depicted as follows: $\kappa = \rho^{-1}$ (m^{-1}). Curvature is unique since every vehicle creates its own curvature when traversing any arbitrary road. However, all roads have already a pre-determined curvature that was designed for a variety of vehicles as discussed before. Furthermore, curvature can be directly related to a vehicle's heading angle through an orthogonal shift for an instantaneous point in time. For this reason, being able to provide vehicles with this pre-determined curvature poses a new guiding factor that autonomous vehicle technology can implement into their decision algorithms for navigation. This in turn can improve their reliability under conditions where disruptions such as weather effects or poor lane markings disable sensor information.

There exist multiple types of roads in which curvature changes as a function of segment length (i.e. $\kappa(s)$) to provide both stability and comfort to drivers. These are divided in two categories, horizontal and vertical curves. Horizontal curves focus on the direction of the centerline, while vertical curves focus on the slope of the centerline. Horizontal curves are divided into 4 main categories shown in Figure 1. For this project, only the 4 main horizontal curves are considered because they constitute most of the implemented roads available.

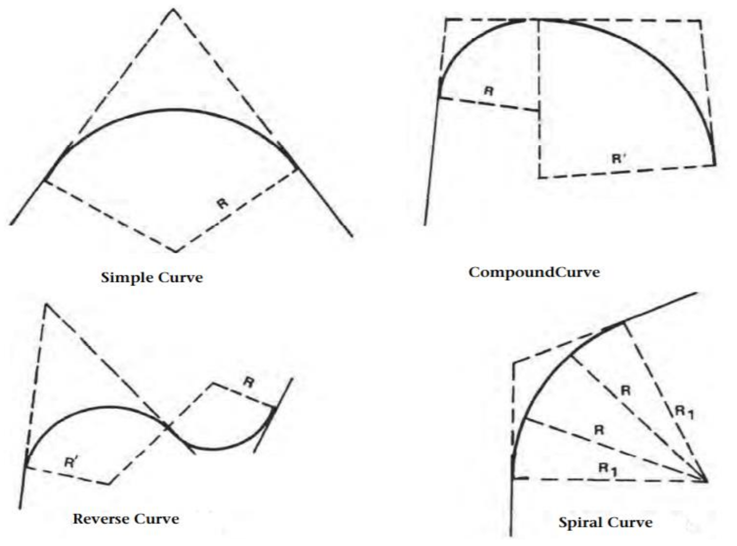

Figure 1. Horizontal Curves

Thus, curvature models were developed that can represent these types of roads. However, to attain an optimized curvature model that can be applied to general vehicles. An optimization problem needs to be defined that utilizes both real road data and vehicle parameters.

Optimization Problem Formulation

To obtain the curvature data, a previous study has been performed which outputs the curvature values needed to be compared with the curvature models [3]. These models are subject to an unconstrained Least Squares Error - Minimization problem such as:

$$\min_{x} \|\kappa_m(s) - \hat{\kappa}\|^2 \quad Pr.1$$

Where:

$\kappa_m(s) =$ Road curvature model m $\qquad$ $\hat{\kappa} =$ Road sampled curvature data

This minimization problem (Pr.1) will test different analytical curvature models and obtain appropriate parameters for each of them accordingly. It is important to note that the models κ_m can be either linear or non-linear.

Pr.1 will focus on representing models κ_m as a representation of any generic road data input. However, to take into consideration the vehicle dynamic stability, Eq.1 and Eq.2 are used as a second minimization problem (Pr.2) that will minimize the steering wheel angle δ (x_1) and vehicle velocity v (x_2) with the following objective function:

$$\min_{x} \; x_1 - \left(57.3L + \overline{UG}x_2^2\right)\kappa_m(s) \qquad Pr.2$$

$subject\ to$:

$$\frac{x_2^2}{g}\kappa_m(s) - \frac{\mu + 0.01e}{1 - 0.01\mu e} = 0$$

This will find the optimized combination for both traveling velocity and wheel angle that uses the model κ_m as part of their process. It is important to note that Pr. 2 has to be solved many times for as many values the vector κ_m, contains. Also, Pr. 2 is a non-linear constrained optimization problem in which the vehicle parameters $(L, \overline{UG})$ and road parameters (μ, e) are regarded as constants for any generic road/vehicle. In general, there exists design vehicle ranges, and steering wheel angles such that extra constraints could be added to Pr. 2 in the following manner:

$$v_{min} < x_2 < v_{max} \quad \& \quad \delta_{min} < x_1 < \delta_{max} \qquad C.1$$

Combining Pr. 2 and C.1 it is possible to form an iterative method in which the velocity and wheel angle of a vehicle, can be optimized with respect to a previously optimized curvature profile. Note: The understeer gradient $\overline{UG}$ is a function of many tire dynamic parameters including velocity and wheel angle. Thus, for this project, it will be regarded as a constant that complies with C.1.

Road Curvature Models

All horizontal curves (in Figure 1) are made of curvatures with constant, and/or linear slopes. To create a model to be optimized in Pr. 1, the following model is proposed:

Piece-wise linear model:

$$\kappa_1 = \left(\frac{x_5}{x_2 - x_1}\right)(s - x_1)[\varphi(s - x_1) - \varphi(s - x_2)] + x_5[\varphi(s - x_2) - \varphi(s - x_3)] + \left(\left(\frac{x_5}{x_4 - x_3}\right)(-s - x_3) + x_5\right)[\varphi(s - x_3) - \varphi(s - x_4)] \quad \text{M.1}$$

Where:

$\varphi(\mathrm{x} - \mathrm{a})$ = Unit Step Function with a-shift

x_i = Variables that describe curvature function with $i \in [1,5]$

M.1 was mathematically designed to be continuous for all s and offers the flexibility of having its x_i parameters to be easily identified as basic geometric properties of a trapezoid. This is illustrated in Figure 2 for the general model κ_1 along s with parameters x_i.

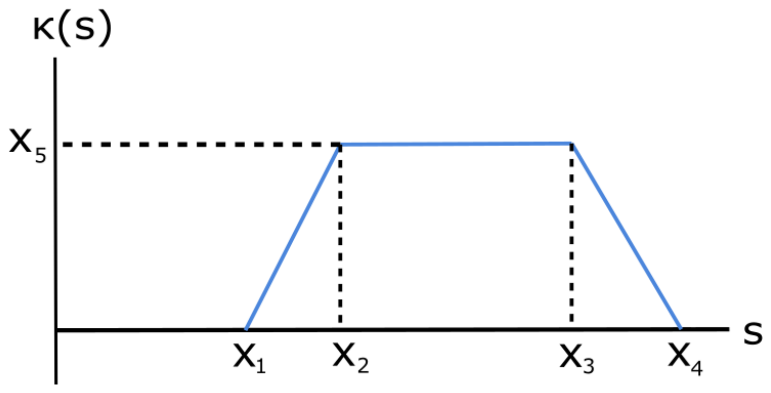

Figure 2. General curvature model 1 (M.1), with designated variables

Note: Mathematical derivations of M.1 are not provided in this report.

Implementation/Results

Least Squares Minimization Pr.1

To optimize Pr. 1, a sample curvature data was generated so that it resembles a compound curve (Figure 1) with Gaussian noise added. The optimization was performed with MATLAB's lsqcurvefit optimization routine (see code in Appendix). It is crucial to note, that for convergence of Pr. 1 with M.1, starting points x_i must be given, and they must not be repeated values so that $x_i \neq x_{i-1}$. A sample optimized M.1 is shown in Figure 3 along with Table 1 with its corresponding coefficients. The behavior of M.1 under many datasets was tested, and it is noticeable how data curves that lack a "downward" slope at the end of the are still being able to be modeled by M.1.

The model can do that by providing a combination of x_3 and x_4 so that the slope of that section is approximately linear.

Table 1. Variables obtained for M.1

Variable	Quantity
x_1	1.543
x_2	14.505
x_3	41.925
x_4	151.046
x_5	0.0261

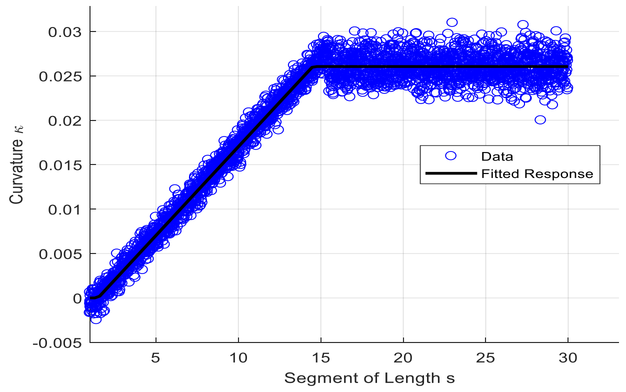

Figure 3. Optimization of Curvature M.1 from Pr. 1

Visualization of Non-linear Constrained Optimization Pr.2

A contour plot of Pr. 2 is shown in Figure 4, with ranges that are realistic for C.1. It is noted that the minimizer of Pr. 2 lies somewhere in the line generated from the intersection of Eq. 1 and Eq. 2. Also, Table 2 provides with the basic road and vehicle parameters that are used throughout Pr. 2. It is important to note that the model is unbounded without the aforementioned constraints.

Table 2. Input Parameters for M.1

Parameter	Quantity	Unit
κ	0.0167	m^{-1}
$\overline{UG}$	1.95	deg
L	2.5	m
g	9.81	m/s^2
e	6	%
μ	0.4	----

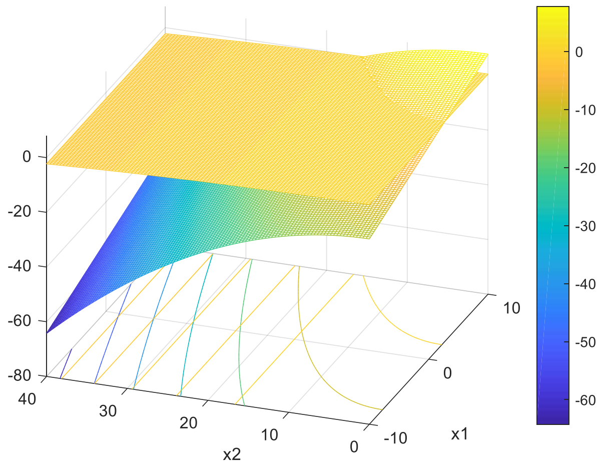

Figure 4. Contour Plots for M.1

AMPL for Optimization of Pr. 2

To minimize Pr. 2, the AMPL modeling language was used, so that complexity of selection for appropriate optimization solvers is reduced [4]. The basic idea of AMPL is to provide 3 files, of the type *.mod*, *.dat*, and *.run*. These are used along with a server that is able to compile all files and provide an optimized solution. The highest versatility of this modeling language is that resembles closely mathematical formulations, such that the transition from math writing to computer input is greatly reduced. Summarizing: *.mod* files contain 4 main components:

`param`: Parameters, which are values can be arbitrarily changed, and are obtained from *.dat* files.

`var`: Variables, which are the optimized variables that need to be found.

`minimize`: The objective equation to be minimized, which should contain all variables, and some (not all) parameters defined before.

`subject to` (or alternatively, **`s.t.`**): Constraints that are applied to the objective function, that can contain some or all variables and parameters previously defined.

In general, all parameters are defined in the *.dat* file, and variables need to be given an initial guess, which is also done in the *.dat* file (see code in Appendix for example). Then, the *.mod* file takes all the data to optimize the model (equations and constraints). Finally, the *.run* file is utilized to simply compile the other 2 files together. It is also used to display specific values or results as deemed appropriate by the user.

Below, Figure 5 is the output obtained from running the files on the Appendix. As it is noticed, this shows the optimized wheel angle and velocity from Pr. 2 subject to C.1 as well. Also, the solver used was MINOS 5.51.

```
ampl: include Problem2.run
minimize Z:
-0.00331957*x[2]^2 + x[1] - 2.24198;

subject to C1:
0.00170234*x[2]^2 = 0.471311;

MINOS 5.51: optimal solution found.
6 iterations, objective -6.161032377
Nonlin evals: obj = 21, grad = 20, constrs = 21, Jac = 20.
Z = -6.16103
x[1] = -3
x[2] = 16.6391
```

Figure 5 AMPL Console Results for Pr. 2

For Pr. 2, the obtained angle is 3 deg, and velocity is 16.63 m/s or about 36 mph, which is a reasonable velocity for the input curvature 0.0167 m^{-1}. Thus, at this curvature segment, the velocity and angle have been optimized. However, this is for a singular curvature data point. Thus, to obtain the ideal velocity and angle profiles, the routine in Pr. 2 needs to be implemented in every segment length and curvature from Pr. 1. A MATLAB pseudo-code is provided where Pr. 1 is solved, and the results are used to iteratively solve Pr. 2.

```
Load Road Data
Calculate Curvature for Road Data $\hat{\kappa}[s]$
Select a Curvature Model M.1
        i.e. $\kappa_1(s)$
Solve Least Squares Optimization of M.1 with Curvature Data $\hat{\kappa}[s]$
        i.e. Pr.1 formulation
To Obtain Optimized Curvature $\kappa_{Opt}(s)$
Define Vehicle/Road Parameters as appropriate
        i.e. friction, gravity
for Step = Initial Step : Total Steps in $\kappa_{Opt}$
    Objective Function
        i.e. Pr.2 formulation
    Define Constraints
        i.e. C.1 Constraints
    Apply Non-Linear Solver to find Optimal driving parameters
        [$v_{Opt}$] = Non-linear-Optimization-Solver(Pr.2,C.1)
end
```

Figure 6. Pr.1 and Pr.2 Solved Iteratively Pseudo-Code

The results from the Pseudo-Code in Figure 6, are shown below in Figure 7. The road data comes from a previous study where an ideal spiral curve was created (Figure 8 left). The solution to Pr.1 was obtained (Figure 8 center) and was utilized to obtain an optimized velocity profile (Figure 8 right). It is important to remark that the road data was created mathematically to be exact with road design standards, thus no noise is present during this implementation. However, it is clearly appreciated how during a spiral curve, the velocity to provide "an optimal" ride changes during the transitions of constant and linear curvatures.

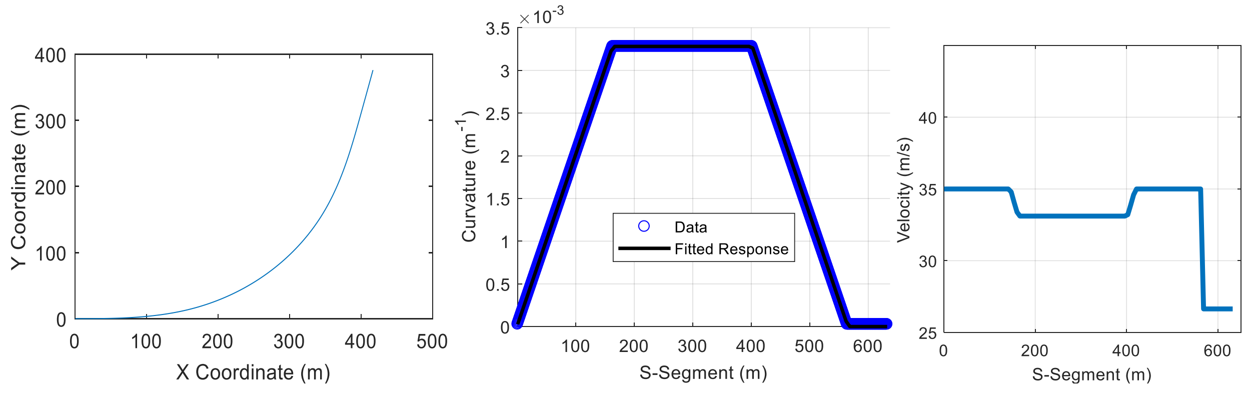

Figure 7. Road Data (left), Curvature Profile (center), and Velocity Profile (right).

Conclusions

In this report, a method to obtain optimized profiles for curvature/velocities/angles was presented and analyzed. The method was constructed based from vehicle dynamics, and road design standards. The optimization tools were MATLAB and AMPL. The method was subdivided in a Least Squares Optimization part (called Pr.1), and an Iterative Nonlinear-Optimization part (called Pr.2). The results for this method show appropriate results for optimal velocity with a spiral curvature road data. In conclusion, the road curvature optimization method shows promising results for optimizing vehicle guidance parameters subject to road/dynamic constraints with general road data input given.

References

1. *A Policy on Geometric Design of Highways and Streets (The Green Book)* Sixth Edition (American Association of State Highway and Transportation Officials, 2011).
2. Gillespie, T.D., *Fundamentals of Vehicle Dynamics* (SAE International, 1992). ISBN:1-56091-199-9.
3. Jacome, R., Stolle, C. and Sweigard, M., "*Road Curvature Decomposition for Autonomous Guidance*," SAE Technical Paper 2020-01-1024, 2020, doi:10.4271/2020-01-1024.
4. Robert Fourer, David M. Gay, and Brian W. Kernighan "*AMPL: A Modeling Language for Mathematical Programming*" Second edition, ISBN 0-534-38809-4
5. Dixit, N.R., *"Evaluation of Vehicle Understeer Gradient Definitions"*. Master's Thesis, 2009.
6. MATLAB reference Least Squares Optimization: *https://www.mathworks.com/help/optim/ug/lsqcurvefit.html Date of Access: 4-10-20*
7. MATLAB reference Non-Linear Constrained Optimization: *https://www.mathworks.com/help/optim/ug/fmincon.html Date of Access: 4-10-20*

8. Appendix

Codes

MATLAB Contours

```
clear all; clc; close all
% From Carsim paper
% At Rho = 30 m -> U = 1.91
% At Rho = 60 m -> U = 1.95
x1 = linspace(-10,10); %Resonable Angles
x2 = linspace(0,40); % Reasonable Speed
Ranges (m/s)
% 40 m/s ~ 90 mph ~ 144 km/hr
L = 2.5; % m
g = 9.81; % m/s^2
%AASHTO values
e = 6; mu = .4;
U = 1.95; K = 1/60;
[X1,X2] = meshgrid(x1,x2);
Z = X1 - (53.7*L+U*X2.^2)*K;
figure; meshc(X1,X2,Z); hold on
Z2 = - X2.^2*K/g + (mu + 0.01*e)/(1-
0.01*mu*e);
meshc(X1,X2,Z2); colorbar;
xlabel('x1'); ylabel('x2')
```

MATLAB M.1 Pr.1 Minimization

```
s = 1:.01:30; n = numel(s)-1;
y1 = (2.*s(1:n/2) - 3)*1e-3;
y2 = 26*ones(1,n/2)*1e-3;
y1o = awgn(y1,25,'measured');
y2o = awgn(y2,25,'measured');
y = [y1o y2o];
x0 = [1 2 3 4 5];

fun2 = @(x,s) ((x(5)./(x(2)-x(1))).*(s -
x(1))).*(heaviside(s-x(1)) - heaviside(s-
x(2))) +...
     x(5).*(heaviside(s-x(2))-heaviside(s-
x(3))) + ...
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+ x(5)
).*(heaviside(s-x(3)) - heaviside(s-x(4)));

x = lsqcurvefit(fun1,x0,s(1:end-1),y)
times = linspace(s(1),s(end-1));
figure
hold on; plot(s(1:end-1),y,'bo')
plot(times,fun2(x,times),'k-','linewidth',2)
legend('Data','Fitted
Response','location','best');
title('Optimization of Curvature Model
\kappa_1'); grid on
xlabel('Segment of Length s');
ylabel('Curvature \kappa');
xlim([times(1), times(end)+5])
```

AMPL Model File: Problem2.mod

```
param L > 0; # m
param K > 0; # m^-1
param U > 0; # deg
param g > 0; # m/s^2
param e > 0; # (Percentage 0-100)
param mu > 0; # unitless

var x{1..2};
minimize Z:  x[1] - (53.7*L + U*x[2]^2/g )*K;

subject to C1:  x[2]^2*K/g - (mu + 0.01*e)/(1-
0.01*mu*e) = 0;
subject to C2: x[1] <= 3;
subject to C3: x[1] >= -3;
subject to C4: x[2] >= 0;
subject to C5: x[2] <= 60;
```

AMPL Data File: Problem2.dat

```
data;

param L := 2.5;
param K := 0.0167;
param U := 1.95;
param g := 9.81;
param e := 6;
param mu := 0.4;

var x:=
       1   1
       2   1;
```

AMPL Data File: Problem2.run

```
#RESET THE AMPL ENVIROMENT
reset;

#LOAD THE MODEL
#model example3.mod;
model Problem2.mod;

#LOAD THE DATA
data Problem2.dat;

#expand Z,C1,C2,C3,C4,C5;
expand Z,C1;

solve;

display Z, x[1], x[2]
```

MATLAB M.1 Pr.1 Minimization with Pr. 2 Optimization Iteratively

```
clear; close all; clc
%Debugging, needs to find correct numbers.
%GPS DATA
%load('CVF9LatX.mat');
load('CVF9LongY.mat');
%Ideal AASHTO
load('MichXm.mat'); load('MichYm.mat');
%x2 = LatX'; y2 = LongY';
x2 = xm'; y2 = ym';
x2 = unique(x2); y2 = unique(y2);
x2 = x2(1:numel(y2));
X = [x2',y2'];
[L,R,K] = curvature(X);
K(1,:) = []; K(end,:) = []; L(1,:) = [];
L(end,:) = [];
x2(1) = []; x2(end) = []; y2(1) = [];
y2(end) = [];
figure; plot(x2,y2);
xlabel('X Coordinate (m)'); ylabel('Y
Coordinate (m)')
title('Raw Road Data')
figure;
h = plot(x2,y2); grid on; axis equal;
set(h,'marker','.');
xlabel('X Coordinate (m)'); ylabel('Y
Coordinate (m)')
title('Road with Curvature Vectors')
hold on
quiver(x2',y2',K(:,1),K(:,2)); hold off
y = sqrt(K(:,1).^2 + K(:,2).^2);
s = L;
% Initial Conditions, NEVER repeat them.
x0 = [100 200 300 400 500];
% Curvature Model M.1
M1 = @(x,s) ((x(5)./(x(2)-x(1))).*(s -
x(1))).*(heaviside(s-x(1)) - heaviside(s-
x(2))) +...
     x(5).*(heaviside(s-x(2))-heaviside(s-
x(3))) + ...
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+ x(5)
).*(heaviside(s-x(3)) - heaviside(s-x(4)));
% Pr.1
fprintf('Pr. 1, Least Squares Min. Has
finalized');
options = optimset('Display','off');
x =
lsqcurvefit(M1,x0,s(1:end),y,[],[],options)
snew = linspace(s(1),s(end),100); % <---
This defines the
% size of the "K_vector".
figure; hold on;
plot(s,y,'bo');
xlabel('S-Segment (m)'); ylabel
('Curvature(m^{-1})');
plot(snew,M1(x,snew),'k-','linewidth',2);
xlim([snew(1), snew(end)+5]);
legend('Data','Fitted
Response','location','best');
title('Data and Fitted Curve'); grid on
% --------------------------
%Parameters
global K_temp e g mu U
% Vehicle Only
L = 2.5;  %U = 1.95;
U = 3;
% Road Only
e = 6; mu = 0.3;
% Both
g = 9.81; K_vector = M1(x,snew);
% --------------------------
%Iterative Optimization Routine for Pr.2
given Optimized M.1
for i = 1:length(K_vector)
K_temp = K_vector(i);
% Objective Function Pr.2
fun = @(x)  x(1) - (53.7*L +
U*x(2)^2/g)*K_temp;
%C.1 (Bounds)
% lb = [-3,25];
% ub = [3,60];
lb = [-3,25]; % -3 < x1 < 3;
ub = [3,35];  % 55 < x2 < 80; mph
% There are no linear constraints, so set
those arguments to |[]|.
A = [];  b = []; % Linear In-equality
Constraints
Aeq = []; beq = [];  % Linear Equality
Constraints
%Initial Conditions
x0 = [1/4,1/4];
%Constraints as an annoynomous function
nonlcon = @EqConstraint;
options =
optimoptions('fmincon','Display','off');
Op(i,:) =
fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,opt
ions);
end
fprintf('Pr. 2 Has finalized \n');
vOpt = Op(:,2);
figure; plot(snew,vOpt,'linewidth',3)
ylim([max(vOpt)-10 max(vOpt)+10])
title('Segment Length vs Velocity
Optimized'); grid on
xlabel('S-Segment (m)'); ylabel ('Velocity
(m/s)');
figure; plot(M1(x,snew),vOpt)
ylim([max(vOpt)-10 max(vOpt)+10])
title('Curvature vs Velocity Optimized');
grid on;
xlabel('S-Segment (m)'); ylabel
('Curvature(m^{-1})');

% Nonlinear Constaints (Not bounds)
function [c,ceq] = EqConstraint(x)
global K_temp e g mu
%Pr.2
% Nonlinear Inequality Constraints
c = x(2)^2*K_temp/g - (mu + 0.01*e)/(1-
0.01*mu*e);
% Nonlinear Equality Constraints
ceq = [];
end
```