Formula SAE Suspension Points Camber Optimization

ME 465 Honors Project
Nic Wiggins
04/24/2020

Pledge: I have neither given nor received any unauthorized assistance on this assignment.

Signature: ______________________ Print Name: _Nicholas C Wiggins________

Introduction

Formula SAE is a collegiate competition with schools from all over the world competing in competitions all over the world. A vast majority of the points in competition come from dynamic events, where the car is driven by student drivers on different courses. Not only do students drive the car, but there is a design portion where students present their designs to engineers in industry. Here students are tested on their understanding of their systems and their reason for design decisions.

The whole point of the suspension system is to optimize the tires contact with the surface. When designing suspension, kinematic analysis is done to produce many curves such as toe change in roll, change in roll centers during suspension travel, and camber change over suspension travel. The most common suspension seen in Formula SAE is a double wishbone suspension. As someone designing suspension points there are 10 different points that need to be placed in space.

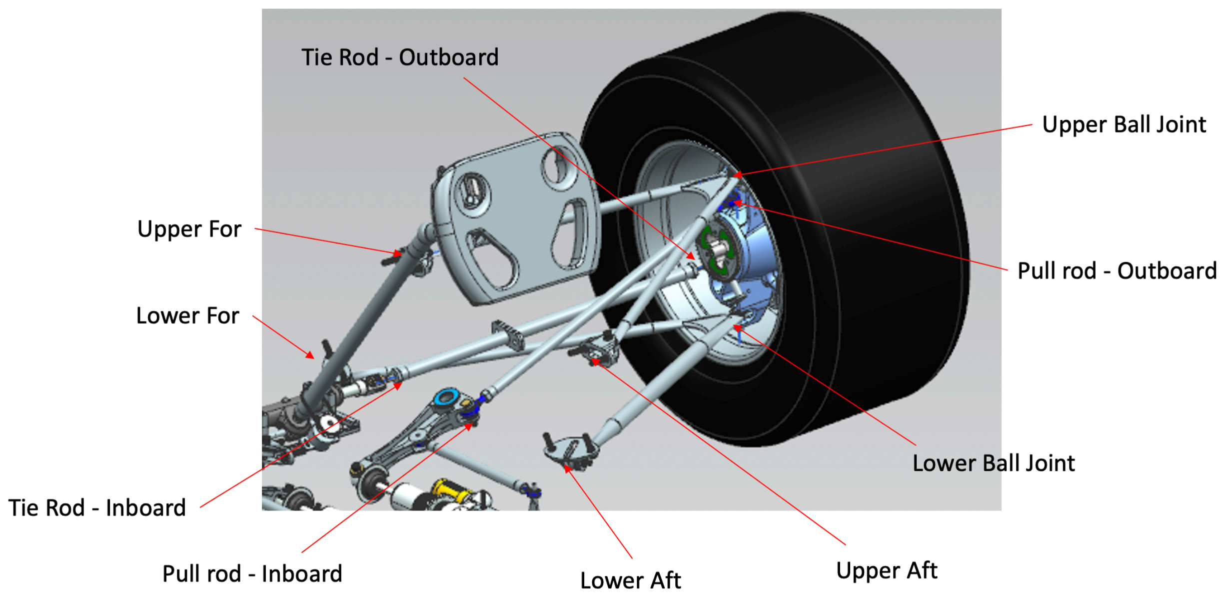


Figure 1. Suspension Points for Double Wishbone Suspension.

The process of picking these points is usually manually adjusting the points in a software that can then sweep through the motion of the suspension and generate the curves. Depending on the results of the curves, one goes back and adjusts these points in space. This becomes a very time intensive iterative process. The purpose of this project was to show the feasibility of designing suspension kinematics with HEEDS for the Formula SAE team. An analysis model was created in MATLAB based off of a research paper, as the motion of the system is fairly complex. HEEDS was used to optimize the points to match a desired camber vs wheel travel curve using root mean square error (RMSE).

Model

The model used was based off the work of Engin Tankik and Volkan Parlanktas from Hacettep University [1]. They created a model to determine the new location of points in a double wishbone suspension for a given input of the lower control arm angle.

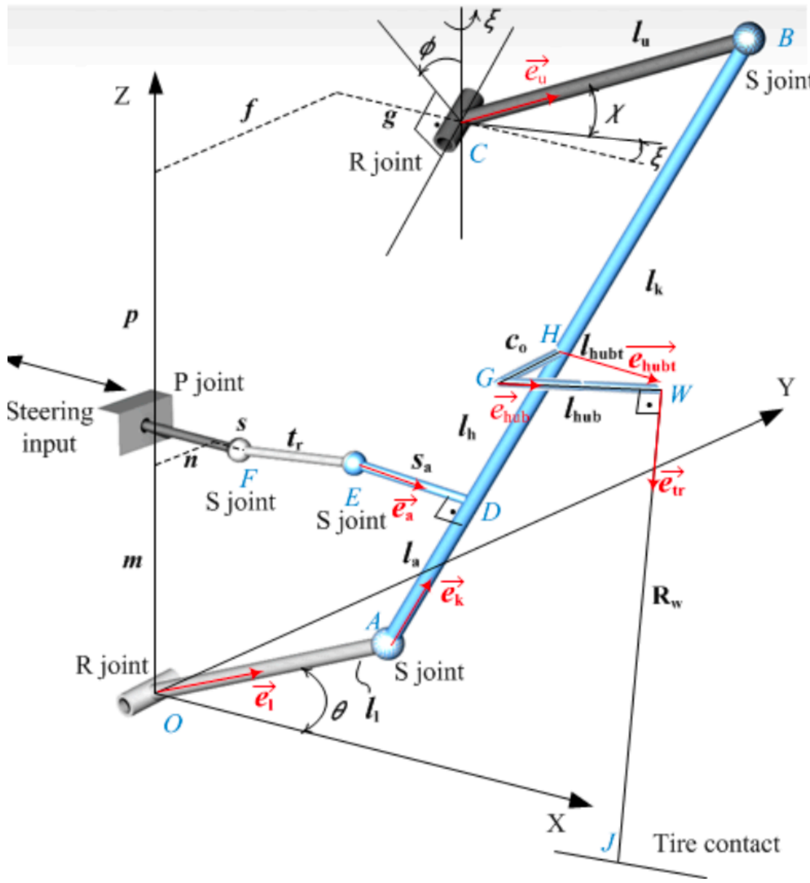


Figure 2. Model Parameters [1].

Their model does have some limitations. Pull rod location is not taken into account, but this shouldn't have too much of an effect on the output curves. The biggest limitation is that it assumes the lower wishbone axis to be parallel with the axis going down the middle of the vehicle. The Formula SAE suspension's lower control arm axis is not parallel with the centerline of the vehicle, however for simplicity sake it was assumed to be this way to allow the use of this model as is. The key variables are defined in Table 1, however a complete list is available in the paper [1]. Using two loop equations they are able to calculate the change in angle χ for a given change in θ, see Figure 2. Then they were able to calculate the new location of the point E, through a system of three nonlinear equations. There was, however, some trouble when using their model. With the example given in the paper, I would get imaginary components for the calculated χ – this is incorrect. After this, I derived my own loop equation and got something different from what was in the paper, however this angle was too small. To solve the equation, I was able to use numeric solving function within MATLAB. Another issue I ran into was solving the system of three nonlinear equations to find the new location of E. To counter this, I tried using numerical solve with no luck. I wrote my own solver for this, but it was not efficient at all, taking up to 10 minutes per simulation. To decrease run time, I ended up keeping the location of E constant for now. This essentially means that both caster and toe travel over the wheel travel will be incorrect, however, this decreased evaluation time down to about 5 seconds.

Table 1. Variable Definitions.

O	Point of centered lower control arm	B	Upper ball joint
A	Lower ball joint	C	Point of centered upper control arm
E	Location of steering arm/tie rod connection	F	Location of steering rack/tie rod con.
sa	Steering arm on upright	tr	Tie rod length
H	Point on AB where wheel is centered	W	Wheel center
c_o	Caster offset	R_W	Effective radius of the tire
ϕ	Anti-Dive Angle	ξ	Yaw Angle of Upper Wishbone
χ	Angular Displacement of Upper Wishbone	l_k	Length of Upright
l_u	Upper control arm length	l_l	Lower control arm length

Posting the code for the model would be too much to put at the end of this report, the project is hosted on GitHub and can be found at [2].

Optimization Setup

Historically when determining suspension points to get desired curves, the points themselves were moved and iterated. Therefore, it would have made sense to allow HEEDs to move these points in 3-D space, similar to what a person would do. For the sake of using this model well, variables were defined off of those seen in Figure 2 and defined in Table 1. This also allowed for straight forward constraint equations. One constraint of the system would be to keep the track width somewhat constant to where it currently is. However, track width changes can be made up for in different designs of the hubs and uprights, therefore, track width changes were essentially kept from getting too far out of control by the bounds in which the core points could move. The biggest constraint is that the upright needs to be able to fit in the wheel shell. To do this, I wanted to make sure the x axis difference between the upper and lower ball joint (B,A) was less than two inches. If more than this, the lower or upper control arms may hit the wheel shell during movement. The system was defined to have 0 initial caster, this means that the center of the wheel is wherever AB intersects Rw at point H. So, naturally this hub location should be less than the upper ball joint in the z direction, and lastly, A and B should fit in the wheel shell when the wheel is centered at H. The left side of the constraints were calculated in MATLAB.

Objective: Minimize RMSE error between projected camber and target camber curve

Constraints:

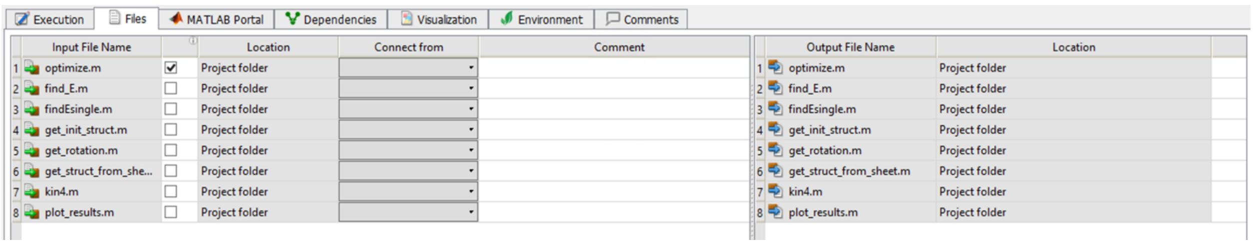
Difference in ball joint x	*ball_joint_delta_x*	$abs(A_x - B_x) - 2 < 0.01$
Ensure upper ball joint fits in wheel shell	*ubj_rad*	$4.5 - \sqrt{\left(B_y - H_y\right)^2 + (B_z - H_z)^2} > 0.01$
Ensure lower ball joint fits in wheel shell	*lbj_rad*	$4.5 - \sqrt{\left(A_y - H_y\right)^2 + (A_z - H_z)^2} > 0.01$
Ensure hub location is below upper ball joint	*Ubj_positive*	$(B_z - H_z) > 0.01$

Design Variables:

Static lower wishbone angle	θ_0	(Continuous)
Length of lower wishbone	l_l	(Continuous)
Static location of steering arm and tie rod joint – x axis	E_x	(Continuous)
Static location of steering arm and tie rod joint – y axis	E_y	(Continuous)
Static location of steering arm and tie rod joint – z axis	E_z	(Continuous)
Static location of upper ball joint – x axis	B_x	(Continuous)
Static location of upper ball joint – y axis	B_y	(Continuous)
Static location of upper ball joint – z axis	B_z	(Continuous)
Location of upper control arm axis – x axis	C_x	(Continuous)
Location of upper control arm axis – y axis	C_y	(Continuous)
Location of upper control arm axis – z axis	C_z	(Continuous)
Anti-Dive angle of upper control arm	ϕ	(Continuous)
Yaw angle of upper control arm	ξ	(Continuous)
Static camber angle	ε_0	(Continuous)

Heeds Setup

Within HEEDS, the process was set to MATLAB and the main file, *optimize.m*, along with all supporting function files were added to the process.



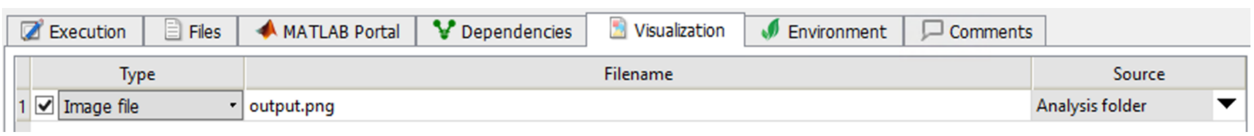
Execution | Files | MATLAB Portal | Dependencies | Visualization | Environment | Comments

	Input File Name		Location	Connect from	Comment
1	optimize.m	☑	Project folder		
2	find_E.m	☐	Project folder		
3	findEsingle.m	☐	Project folder		
4	get_init_struct.m	☐	Project folder		
5	get_rotation.m	☐	Project folder		
6	get_struct_from_she...	☐	Project folder		
7	kin4.m	☐	Project folder		
8	plot_results.m	☐	Project folder		

	Output File Name	Location
1	optimize.m	Project folder
2	find_E.m	Project folder
3	findEsingle.m	Project folder
4	get_init_struct.m	Project folder
5	get_rotation.m	Project folder
6	get_struct_from_sheet.m	Project folder
7	kin4.m	Project folder
8	plot_results.m	Project folder

Figure 3. Process Files.

Since the outputs that are being monitored are all curves, each run would generate and save an image of the core plots. This image was added to the visualization of the process.



Execution | Files | MATLAB Portal | Dependencies | Visualization | Environment | Comments

	Type	Filename	Source
1	☑ Image file	output.png	Analysis folder

Figure 4. Visualization File.

All variables were defined and tagged in *optimize.m*, with the bounds displayed in Figure 5. The baseline was set to the current design of the suspension. The bounds selected for position variables were within 4 inches of the current point location. The idea behind this was to keep the new design in the same ballpark as the current system. Clevises that attach control arms to the chassis could be redesigned, but the chassis remains constant for typically three years. This is why there are tighter bounds on point C, since this involves the chassis. The upright, effectively AB, is redesigned every year, which allows larger bounds on locations of B, as well as the length of the lower control arm, l_l. The steering arm is a part of the upright, which is also adjustable with a clevis – providing large bounds for the location of point E. Static theta angle was allowed to vary from -45 to 45 degrees, since anything beyond this would be unstable. Yaw angle was somewhat limited in its range due to potentially running into the chassis if too large. Anti-dive angle bound was set to 45 degrees, anything larger could also lead to instability. Lastly, initial camber angle was varied in the negative direction since the tires that we run can be optimal under small negative camber values. The responses were tagged and defined as shown in Figure 6. Before the toe output was realized to be inaccurate the initial plan was to minimize error between both desired camber and desired toe curves. As optimizing began, different desired camber curves were tested, hence the reason for multiple *RMSE_camber* responses.

	Variable Name	Type	Min	Baseline	Max	Resolution
1	ex	Continuous	12	16.3048	20	101
2	ey	Continuous	0	2.225	5	101
3	ez	Continuous	-2	1.4101	4	101
4	bx	Continuous	12	15.4548	18	101
5	by	Continuous	-1	-0.375	1	101
6	bz	Continuous	4	6.7601	9	101
7	cx	Continuous	0	3.0008	6	101
8	cy	Continuous	-1	0.245	1	101
9	cz	Continuous	3	5.4351	8	101
10	ll	Continuous	10	15.98	20	101
11	theta_0	Continuous	-45	-0.3941	45	101
12	phi_deg	Continuous	0	1.2519	45	101
13	zeta_deg	Continuous	0	0	20	101
14	camber_deg	Continuous	-2	-0.2	0	101

Figure 5. Variable Definitions.

	Response Name	Source	Formula
1	wheel_travel	Tag	
2	track_variation	Tag	
3	camber	Tag	
4	KPI	Tag	
5	caster	Tag	
6	toe	Tag	
7	ball_joint_delta_x	Tag	
8	lbj_rad	Tag	
9	ubj_rad	Tag	
10	ubj_positive	Tag	
11	RMSE_camber	Curve Fit	RMS(wheel_travel, camber, camber_target, Match, false)
12	RMSE_toe	Curve Fit	RMS(wheel_travel, toe, toe_target, Match, false)
13	TOTAL_RMSE	Formula	RMSE_camber+RMSE_toe
14	RMSE_camber_2	Curve Fit	RMS(wheel_travel, camber, camber_target_2, Match, false)
15	RMSE_camber_3	Curve Fit	RMS(wheel_travel, camber, camber_target_3, Match, false)
16	RSME_camber_4	Curve Fit	RMS(wheel_travel, camber, camber_target_4, Match, false)
17	RMSE_camber_5	Curve Fit	RMS(wheel_travel, camber, camber_target_5, Match, false)

Figure 6. Response Definitions.

HEEDs Study

An initial study was done with a desired camber curve. The optimal curve is seen in Figure 8, or the red curve in 7(camber). This curve was picked since it was somewhat close to the baseline curve, but unique enough to see how HEEDs would change variables to adjust to the desired curve. HEEDs was able to decrease RMSE from 1.153 to 0.145.

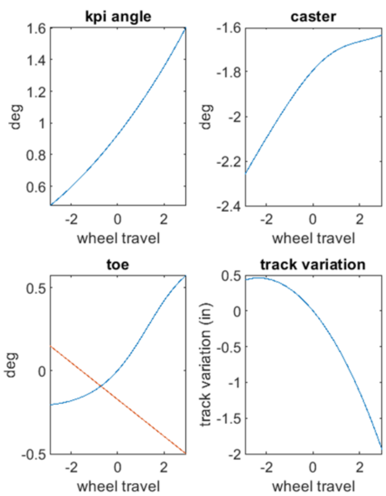


Figure 7a. Baseline Plot.

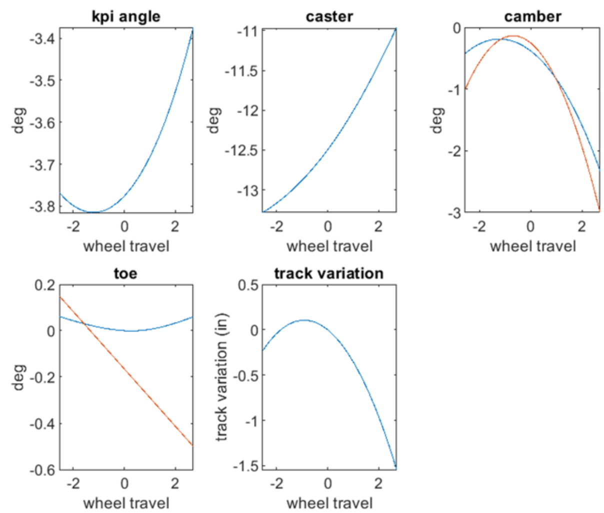


Figure 7b. Optimal Plot.

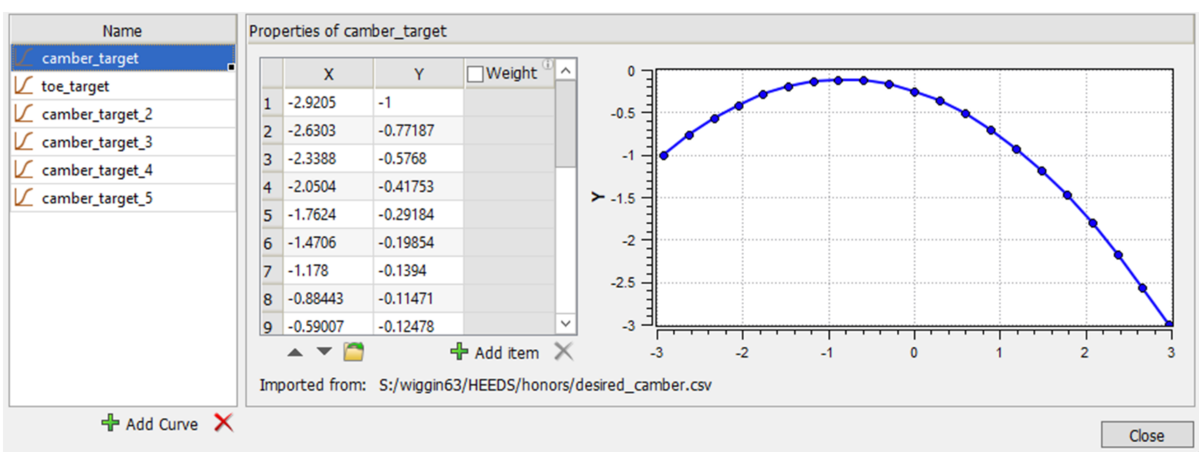


Figure 8. Optimal Camber Curve 1.

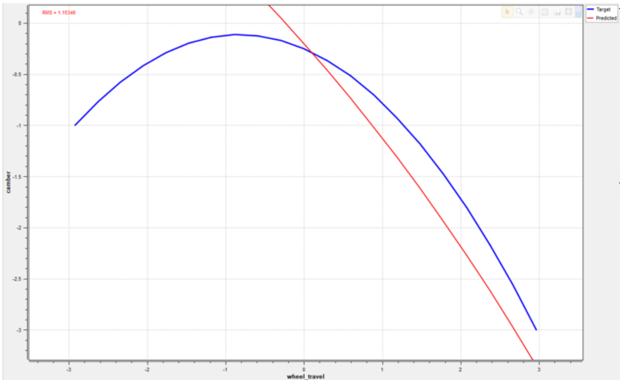


Figure 9a. Initial RMSE.

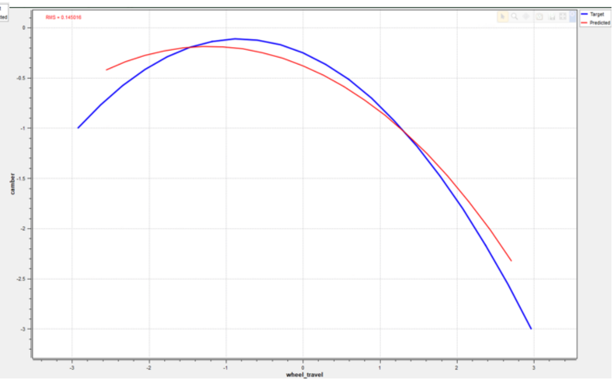


Figure 9b. Optimal RMSE.

Name	Best Value	Change
ex	15.76	-3%
ey	1.6	-28%
ez	2.14	52%
bx	16.44	6%
by	-0.12	68%
bz	6.5	-4%
cx	5.94	98%
cy	0.4	63%
cz	5.5	1%
ll	14.6	-9%
theta_0	2.7	785%
phi_deg	2.25	80%
zeta_deg	5.2	n/a
camber_deg	-0.38	-90%

Figure 10. Variable Changes.

To do this, essentially the upper ball joint was moved down, and further away from the lower ball joint toward the front of the car. HEEDs also moved the upper control arm out away from the chassis, and further forward relative to the car. Anti-dive was increased and yaw angle as well of the upper control arm. Initial camber was also increased in the negative direction. Lastly, the lower control arm got shortened, and its initial position is a slightly larger angle.

While HEEDs ability to bring the RMSE down for this design was impressive, it isn't the desired camber curve for the tires running on the Formula SAE car. Figure 11 shows the FY-SA curve for varying nominal loads and inclination angles. Inclination angle (IA) is directly corelated to the camber on the tire. This indicates that there are higher lateral loads for 0 degrees of IA when there is a large slip angle, and higher lateral loads for 2 to 4 degrees of IA when there is small slip angle. Higher nominal loads and slip angles mean that corner of the vehicle is highly loaded. In this case, the chassis would move down with respect to the wheels, equivalent to positive wheel travel if the chassis were to stay constant.

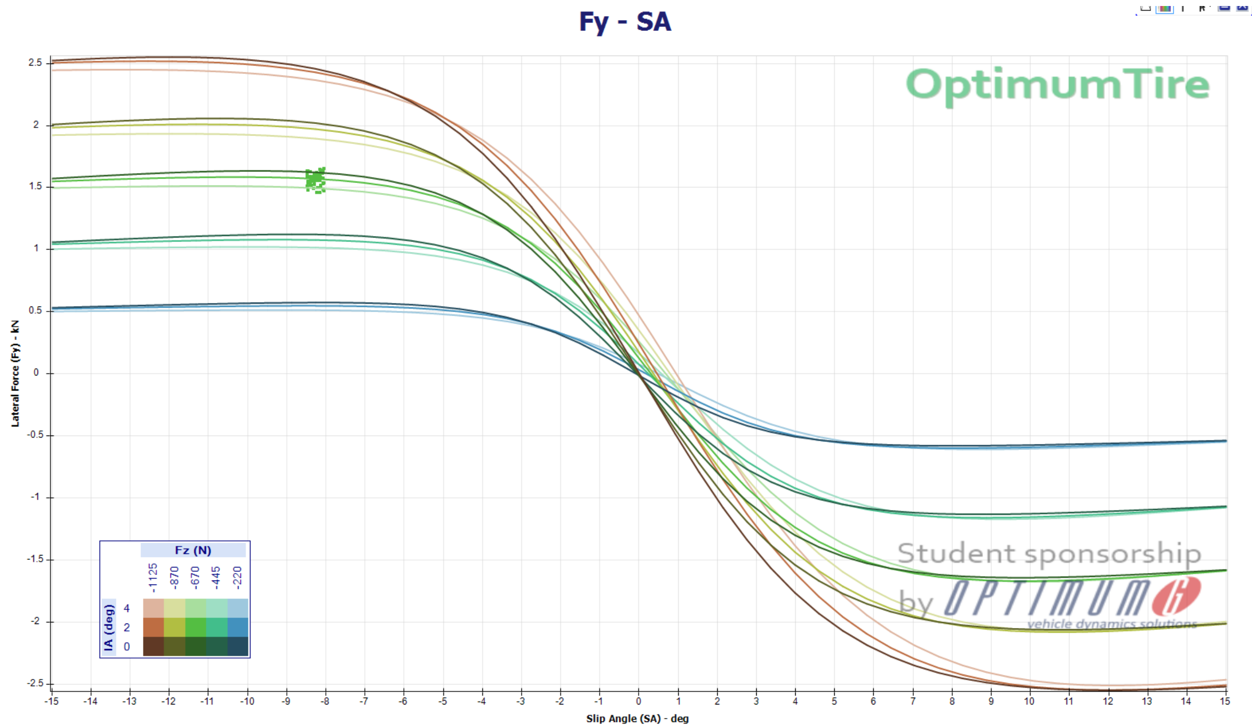


Figure 11. Lateral Force vs Slip Angle for Varying Inclination Angles.

What this means for a desired camber curve is what is seen in Figure 12. Ideally it would be flat instead of overshooting 0 near the bounds of wheel travel, but for sake of easy curve generation using a polynomial fit in MATLAB, this curve was used. Note, at 0 wheel travel the desired camber was set to -2 not -4 degrees, since -4 degrees has the potential to decrease tire life significantly.

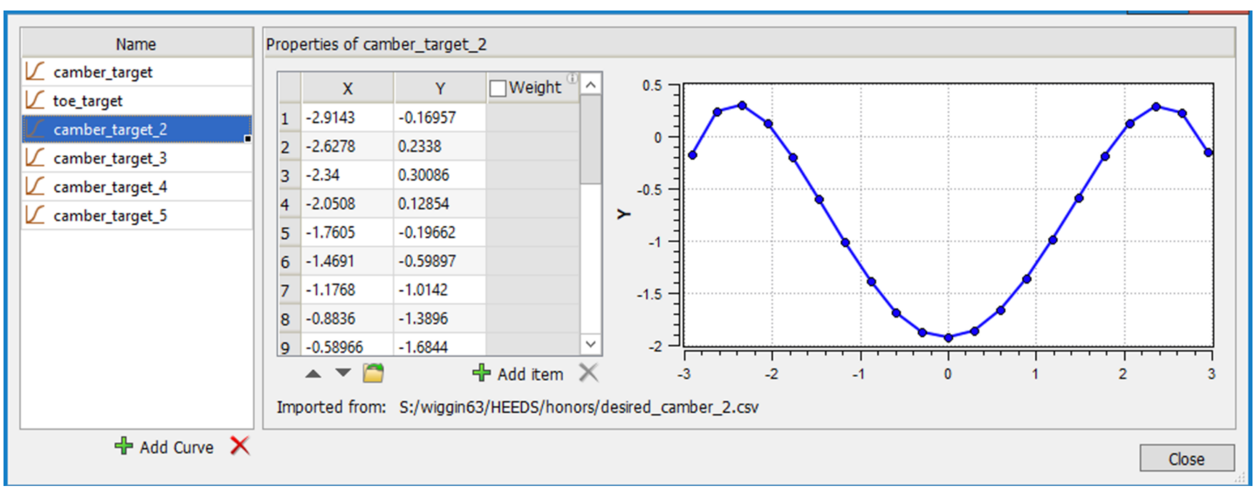


Figure 12. New Ideal Camber Curve.

For this new desired curve, 150 iterations were completed, and the results are as follows in Figure 13. While the curve in Figure 13b doesn't match the desired curve exactly, RSME is only decreased from 1.856 to 1.032, its actually pretty good in terms of response.

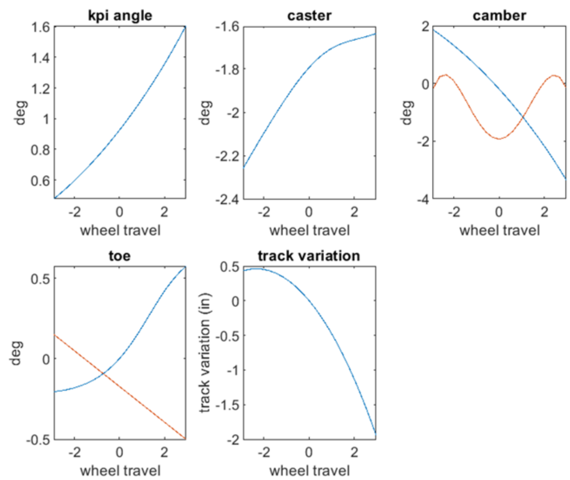


Figure 13a. Initial Plot.

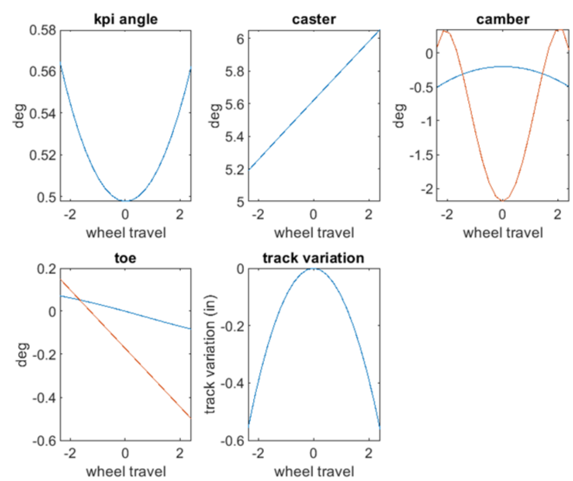


Figure 13b. Optimal Plot.

The camber varies between -0.5 and -0.2 degrees of camber. While this doesn't follow the desired concavity, the values of camber at large wheel travel are relatively small, which as seen in Figure 11, is desired. For most of the dynamic events, the tire would be operating at large slip angles. Therefore, the loss in lateral force at light loading of the tire since camber is not -2 is made up for the gain in lateral force when the corner is more heavily loaded and has a small camber of -0.5.

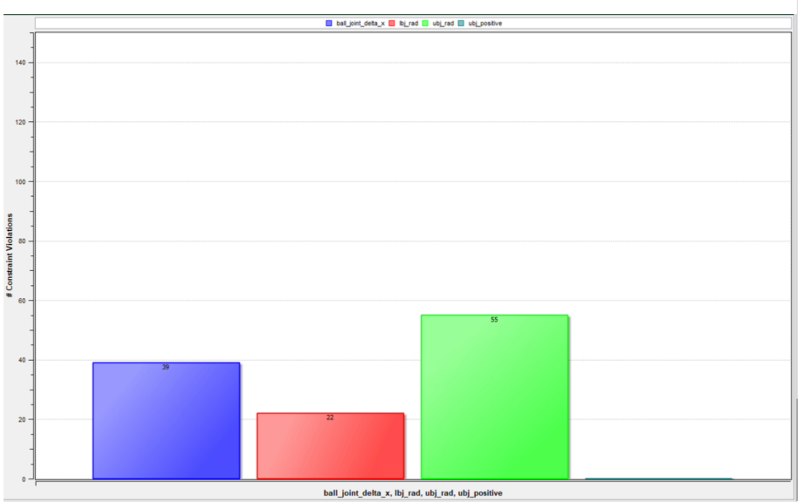


Figure 14. Constraint Violations.

Name	Best Value	Change
ex	18.24	12%
ey	0.55	-75%
ez	1.3	-8%
bx	13.44	-13%
by	0.32	185%
bz	7.75	15%
cx	1.98	-34%
cy	-0.76	-410%
cz	7.7	42%
ll	13.8	-14%
theta_0	0	100%
phi_deg	1.35	8%
zeta_deg	0	n/a
camber_deg	-0.2	0%

Figure 15. Variable Changes.

The optimal design shifted the tie rod/steering arm point toward the rear of the car. It also moved the upper ball joint toward the front of the car, and up a little. The center of the upper control arm axis was also moved toward the read of the car by quite a bit, up and a little closer to the chassis. The lower control arm length was decreased, and the static theta was decreased slightly. However, Anti-Dive and yaw angle of the upper control arm axis remained nearly the same.

The majority of the constraint violations was the upper ball joint distance to center of the wheel was larger than the wheel shell. The lower ball joint distance had 22 violations, and the distance between the upper and lower ball joint x value was larger than 2 inches 39 times. When looking further into these constraint violations using a 2-D relation plot, Figure 16, it was noted that there were multiple designs just to the left of the constraint boundary for the upper ball joint constraint. While its noted the optimal design was not on the constraint boundary, indicating a non-active constraint, the closeness of good designs to the left of the boundary caused a further look into the defined constraints.

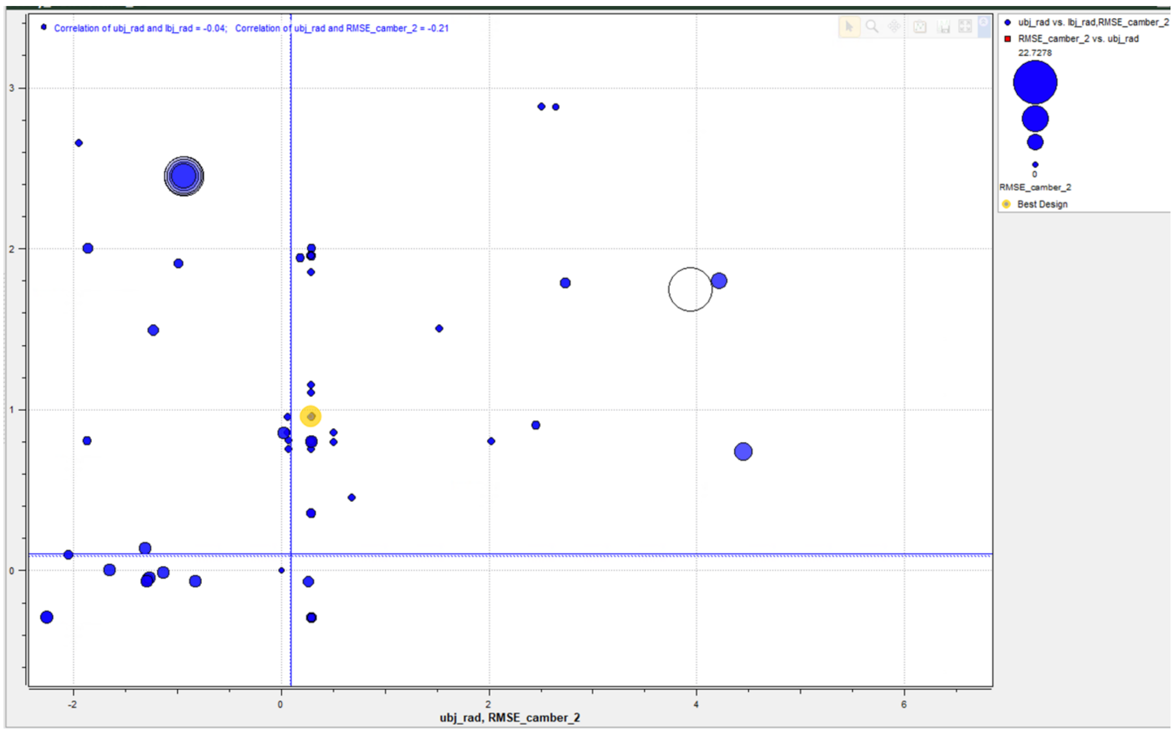


Figure 16. 2-D Relation Plot.

It was noted that the constraints initially, listed in the Optimization Setup, were set to a bound of 0.1. This was set instead of 0 due to the HEEDs error “Normalized value cannot be zero. If using the constraint limit, specify the normalization factor separately”. But there was no need for this to 0.1 rather than zero, since the inner radius of 4.5 inches already included a safety factor. Therefore, a new study was run with each constraint bound set to 0.01, as seen in the Optimization setup. Also, the optimal run for this study was on design 150, so the number of evaluations was bumped to 200 for the new study. With 200 evaluations and the change in constraints, the following camber curve was generated. The RMSE was brought from 1.856 to 0.290.

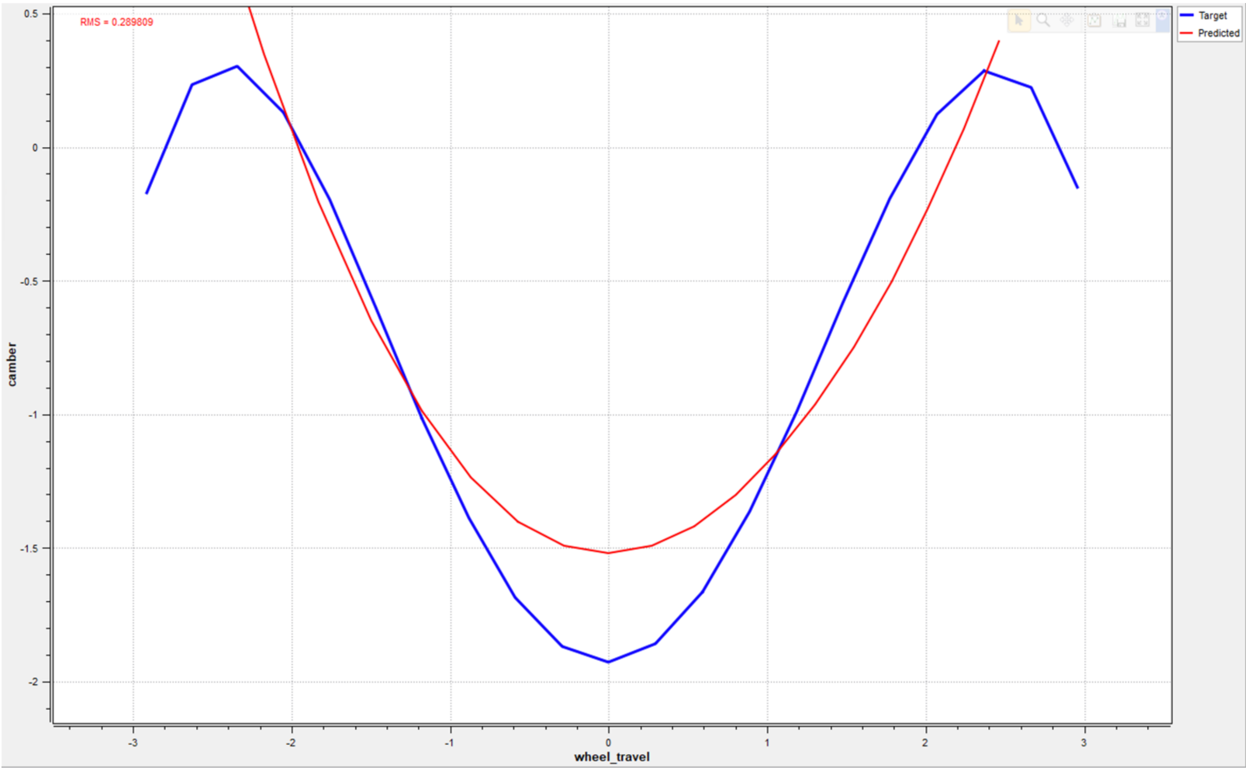


Figure 17. Final Camber Curve.

Name	Best Value	Change
ex	18.88	16%
ey	2.55	15%
ez	-1.58	-212%
bx	16.2	5%
by	-0.46	-23%
bz	7.7	14%
cx	3.78	26%
cy	-0.92	-476%
cz	6.25	15%
ll	15.8	-1%
theta_0	3.6	1,013%
phi_deg	38.25	2,955%
zeta_deg	10.2	n/a
camber_deg	-1.52	-660%

Figure 18. Final Variables.

Despite the changes in the constraint boundaries, this new optimal design didn't actually take advantage of the new boundaries. This design moved the tie rod/steering arm point down by 212%. The upper ball joint was moved toward the rear of the car slightly, and up a little. The upper control arm center was mainly moved toward the rear of the car as well. The length of the lower control arm is about the same, however the static value of the lower control arm angle was increased, and a large amount of anti-dive and yaw angle of the upper control arm axis. Initial camber was also decreased quite a bit, as expected to lower the RMSE.

Conclusion

Overall, it was encouraging to see how HEEDs was able to optimize this very complex system. What would normally take multiple days of someone iterating through different points in space to get the desired responses was now done in about 20 minutes. With these results, it suggests with enough iterations most desired curves are possible. From here, the hope is to develop a new model that would be able to accurately calculate toe angle. In addition, this the model could expand from one corner to all four corners, and parameters like roll center curves could be optimized for. Really, besides model complexity, there is no reason HEEDs could not optimally design the whole suspension kinematics. In this case, the objective would change to minimize the total RSME or it could be a multi-objective search, and weights could be changed depending on the importance of that response.

References

[1] TANIK, Engin, and Volkan PARLAKTAŞ. "On the Analysis of Double Wishbone Suspension." *Hacettepe University*, Journal of Advanced Mechanical Design, Systems, and Manufacturing, 2015, pp. 1–10.

[2] Wiggins, Nic. "Double Wishbone Kinematics." *Github*, Nicwigs, 2020, github.com/nicwigs/465.