Drift Initiation via Nonlinear Trajectory Optimization

John Alsterda, Qizhan Tam

***Abstract*— We study the problem of transitioning an automated vehicle from a normal driving state into a drift equilibrium, a maneuver known to professional rally-car drivers as the 'Scandinavian Flick.' We formulate the problem as a nonlinear control optimization to compute an offline state and input trajectory. Specifically, we use a direct method to formulate a free-time nonlinear program which we solve using IPOPT, an interior point method. Finally, we analyze our control strategy and plan to implement it on MARTY, Stanford's self-drifting DeLorean test platform.**

I. Introduction

Prior research on automated drifting has been presented by J. Goh et al., who demonstrated autonomous tracking of stable drift equilibrium states [1]. They did not, however, address the problem of transitioning a vehicle into drift equilibria from a normal driving state. The goal of this project is to compute optimal control strategies to accomplish such a transition using a direct optimization method and nonlinear programming. The maneuver we focus on is the Scandinavian Flick, which is characterized by saturation (oversteering) of the rear wheels while inertia carries it through a corner. The maneuver was studied by M. Acosta et al., who present an insightful phase-plane analysis of the dynamics and offer a finite state-machine approach to execution [2]. Furthermore, E. Velenis et al. presented a nonlinear optimization strategy to compute trajectories for a related maneuver, the *Pendulum Turn*; we employ a similar method [3].

II. Vehicle Models

A. Single-track Bicycle Model

To formulate our problem and simulate computed trajectories, we employ a simplified vehicle dynamics model shown in Fig. 1, where each of the front and rear wheel pairs are coupled and assumed to have identical forces acting upon them. The position state variables of this model are the car's East-North location and heading angle: $\mathbf{x} = (E, N, \Psi)$; its velocity state is composed of translational and angular velocities: $\mathbf{v} = (U_x, U_y, r)$. The input variables are the steering angle and the torque applied to the rear axle, $\mathbf{u} = (\delta, \tau_\mathrm{r})$. Additional lifting variables that describe the tire forces, $\mathbf{F} = (F_{x\mathrm{f}}, F_{x\mathrm{r}}, F_{y\mathrm{f}}, F_{y\mathrm{r}})$ and tire slip angles, $\boldsymbol{\alpha} = (\alpha_\mathrm{f}, \alpha_\mathrm{r})$ are used to simplify the process of defining dynamic and constraint functions. The vehicle dynamics equations follow:

$$\dot{\Psi} = r$$
$$\dot{E} = -U_x \cdot \sin(\Psi) - U_y \cdot \cos(\Psi)$$
$$\dot{N} = U_x \cdot \cos(\Psi) - U_y \cdot \sin(\Psi)$$

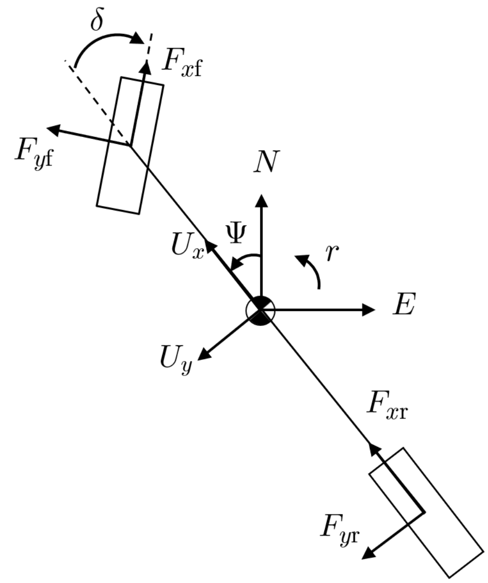

Fig. 1. Variables and parameters of the single-track model

$$\dot{r} = \left(a \cdot F_{y\mathrm{f}} \cdot \cos(\delta) + a \cdot F_{x\mathrm{f}} \cdot \sin(\delta) - b \cdot F_{y\mathrm{r}}\right) I_z^{-1}$$
$$\dot{U}_x = \left(F_{x\mathrm{f}} \cdot \cos(\delta) - F_{y\mathrm{f}} \cdot \sin(\delta) + F_{x\mathrm{r}} + m \cdot r \cdot U_y\right) m^{-1}$$
$$\dot{U}_y = \left(F_{x\mathrm{f}} \cdot \sin(\delta) + F_{y\mathrm{f}} \cdot \cos(\delta) + F_{y\mathrm{r}} - m \cdot r \cdot U_x\right) m^{-1}$$

B. Tire Force Models

The tire forces on the vehicle are governed by the following equations and inequality constraints:

$$F_{z\mathrm{f}} = (m \cdot b \cdot g - h \cdot F_{x\mathrm{r}})/(a+b)$$
$$F_{z\mathrm{r}} = (m \cdot a \cdot g + h \cdot F_{x\mathrm{r}})/(a+b)$$
$$F_{x\mathrm{f}} = 0$$
$$F_{x\mathrm{r}} = \frac{\tau_\mathrm{r}}{R_\mathrm{wr}}$$
$$|F_{x\mathrm{r}}| \leq \mu_\mathrm{r} \cdot F_{z\mathrm{r}} \cdot \cos(\alpha_\mathrm{r})$$
$$\tan(\alpha_\mathrm{r}) \cdot U_x = U_y - b \cdot r$$
$$\tan(\alpha_\mathrm{f} + \delta) \cdot U_x = U_y + a \cdot r$$
$$(\xi \cdot \mu_r \cdot F_{zr})^2 = (\mu_\mathrm{r} \cdot F_{z\mathrm{r}})^2 - F_{x\mathrm{r}}^2$$

For the lateral tire forces $F_{y\mathrm{f}}$ and $F_{y\mathrm{r}}$, we employ a logit approximation of the piecewise Fiala brush tire model used in [1] to obtain a continuous tire model:

$$F_{y\mathrm{f}} = \mu_\mathrm{f} \cdot F_{z\mathrm{f}} \cdot \frac{1 - \exp(w_\mathrm{f} \cdot \alpha_\mathrm{f})}{1 + \exp(w_\mathrm{f} \cdot \alpha_\mathrm{f})}$$
$$F_{y\mathrm{r}} = \mu_\mathrm{r} \cdot F_{z\mathrm{r}} \cdot \xi \cdot \frac{1 - \exp(w_\mathrm{r} \cdot \alpha_\mathrm{r})}{1 + \exp(w_\mathrm{r} \cdot \alpha_\mathrm{r})}$$

We surmised that a continuous model may yield faster and more consistent convergence than a piecewise one (a hypothesis not tested). [2] and [3] use the continuous Pacejka *Magic Formula* to model their tire forces, which requires over a dozen parameters. The large parameter set obscures physical basis for the aptly named *Magic Formula*, and the task of fitting these values presents a hurdle to real-world implementation. Fig. 2 illustrates the similarity between a logit function, the Fiala model, and experimentally collected data. We are aware of no prior work employing a 'logit' tire model.

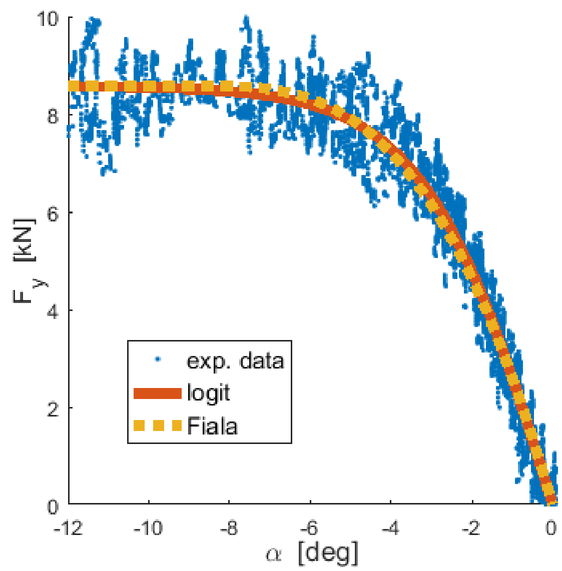

Fig. 2. Comparison among experimental, logit, Fiala tire forces

Input variables are constrained to the following sets:

$$\tau_{\mathrm{r}} \in \left[\tau_{\min}, \tau_{\max}\right]$$
$$\delta \in \left[-\delta_{\max}, \delta_{\max}\right]$$
$$\dot{\delta} \in \left[-\dot{\delta}_{\max}, \dot{\delta}_{\max}\right]$$

III. Problem Formulation

A. State and Input Variables

As detailed above, the state and input variables are $\mathbf{z}(t) = (\mathbf{x}, \mathbf{v}) = (N, E, \Psi, U_x, U_y, r)$ and $\mathbf{u}(t) = (\delta, \tau_{\mathrm{r}})$. These variables are constrained or penalized at $t = 0$ and $t = T$ (as depicted in Fig. 3) in the following way to enforce a transition from normal driving to drifting:

$\mathbf{z}(0)$ is constrained to $(0, -20\,m, 0, 10\,m/s, 0, 0)$, whose values were chosen by intuition, and trial & error, and to represent a normal driving velocity. To determine $\mathbf{z}(T)$ we computed a drift equilibrium, defined by $\dot{r} = \dot{U}_x = \dot{U}_y = 0$. After specifying the desired drift radius $R = 6\,m$ and side-slip angle $\beta = \tan(U_y/U_x) = -40°$, we use the equations of motion to compute the unstable equilibrium

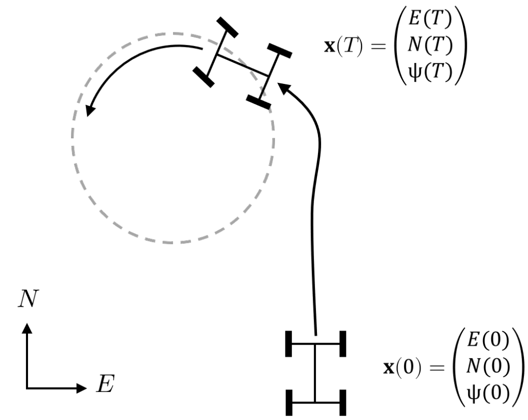

Fig. 3. Desired initial and final positions of the vehicle

$\mathbf{z}(T) \approx (0, 0, 0.7\,rad, 5.2\,m/s, -4.5\,m/s, 1.3\,rad/s)$. To afford the optimization flexibility around the final state, we only constrain the final velocity states. The final position on the other hand, is enforced with a quadratic cost in the objective function. This soft formulation lead to increased robustness in convergence with a negligible effect on computed trajectories.

B. Nonlinear Optimization Problem

The time-optimal drift initiation problem is then:

$$\min_{u(\cdot)} \int_0^T \mathrm{d}t + \|\mathbf{z}(T) - \mathbf{z}_{\mathrm{eq}}\|^2$$

subject to the dynamics detailed in the previous section and the equilibrium constraint:

$$\dot{U}_x(T) = \dot{U}_y(T) = \dot{r}(T) = 0$$

C. Euler Forward Discretization

We discretized time into N stages, where the difference between time steps dt and the final time T are free parameters:

$$t_0 < t_1 < ... < t_N := T$$

The zero-order hold input and state dynamics are then discretized according to $t \in [t_i, t_{i+1}], \forall i \in \{0, 1..., N-1\}$:

$$\mathbf{u}(t) = \mathbf{q}_i$$
$$\dot{\mathbf{x}}(t_i) = \mathbf{a}(\mathbf{x}(t_i), \boldsymbol{l}(t_i), \mathbf{q}_i, t_i)$$
$$\mathbf{x}(t_{i+1}) = \mathbf{x}(t_i) + \mathbf{a}(\mathbf{x}(t_i), \boldsymbol{l}(t_i), \mathbf{q}_i, t_i) \cdot (t_{i+1} - t_i)$$

By letting $\mathbf{g} \leq 0$ and $\mathbf{h} = 0$ represent the inequality and equality constraints, we can summarize the discrete nonlinear problem as follows:

$$\min_{\mathbf{q}_i} \sum_{i=0}^{N-1} (t_{i+1} - t_i) + ||\mathbf{z}(t_f) - \mathbf{z}_{\mathrm{eq}}||^2$$

s.t. $\forall\, i \in \{0, 1..., N-1\}$,

$$\mathbf{x}(t_{i+1}) = \mathbf{x}(t_i) + \mathbf{a}(\mathbf{x}(t_i), \boldsymbol{l}(t_i), \mathbf{q}_i, t_i) \cdot (t_{i+1} - t_i)$$
$$\mathbf{g}(\mathbf{x}(\mathbf{t_i}), \mathbf{l}(\mathbf{t_i}), \mathbf{q_i}, \mathbf{t_i}) \leq 0$$
$$\mathbf{h}(\mathbf{x}(\mathbf{t_i}), \mathbf{l}(\mathbf{t_i}), \mathbf{q_i}, \mathbf{t_i}) = 0$$

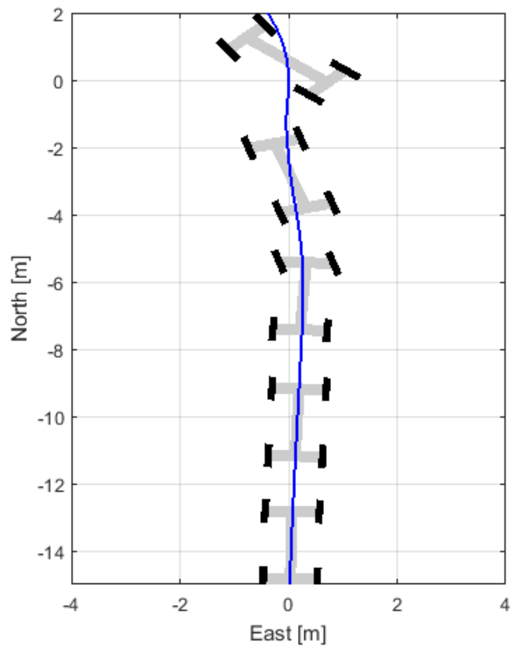

Fig. 4. Snapshot illustration of drift initiation trajectory

The resulting nonlinear optimization is then solved offline using a direct method: IPOPT's interior point nonlinear program, parsed in MATLAB with YALMIP [4][5].

IV. Results

The following section provides an analysis of the drift initiation trajectory calculated by our optimization program. To provide a cursory understanding, please visit the following link to our animated results: `https://youtu.be/JB334BDw6gY`. Note: the drift maintained after $t \gtrapprox 2\,sec$ is presented for visual aid only, and outside the scope of our optimization.

A. Optimal Trajectory Analysis

First we present a qualitative interpretation of the optimized trajectory to provide a physical basis and intuition for the result. The position states we describe are illustrated in Fig. 4; note: the East axis is stretched to exaggerate subtle lateral behavior. The inputs and velocity states are found in Fig. 5, where variables are plotted in blue and drift equilibrium targets are shown in orange. A more in-depth analysis of the role of weight transfer and tire-force limits is found next in section, IV. B.

The maneuver begins at $t = 0\,sec$, when the solution immediately commands the vehicle to accelerate forward, raising longitudinal speed in order to minimize time. Shortly thereafter at $t \approx 0.1\,sec$, the vehicle steers slightly right and away from the turn. This counter-intuitive behavior is consistent with the professional maneuvers described by Acosta et al.! They report the side-slip generated here provides important an yaw moment as the corner approaches.

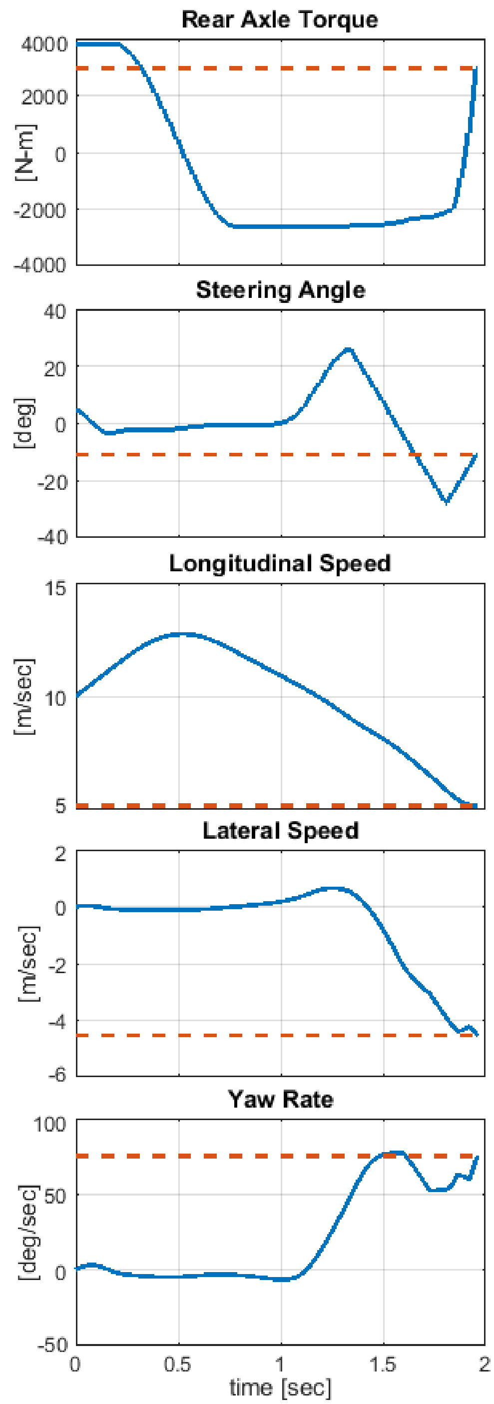

Fig. 5. Optimal input and velocity trajectory

Next, the vehicle steps hard on the brakes, decreasing longitudinal speed toward the target equilibrium speed. Passing the halfway point, steering angle is abruptly commanded hard left which yaws the vehicle significantly! As yaw rate approaches equilibrium, the vehicle *counter-steers* back right with maximum slew rate, which appears to pull lateral speed sharply to the right. Finally, both steering and torque are commanded to their equilibrium positions in tandem, stabilizing the vehicle about a saddle-point equilibrium. Incredible!

B. Weight Transfer Analysis

We are simply amazed how effectively nonlinear optimization and simple models worked to generate this dynamic maneuver — typically reserved for the most exciting professional arenas. One modelling choice in particular yielded insight worth examination: longitudinal weight transfer. This model allows the vehicle weight to shift forward onto the front axle during heavy braking, increasing F_{zf} and thus the amount of frictional force which can generated at the loaded axle (and vise-versa for throttle).

In Fig. 6, the top plots concern force generated at the front axle, while the bottom two illustrate force at the rear. Plots on the left portray the tires' *friction circles*, or static force limits in red. The circle represents the maximum force a tire can generate, without considering weight transfer. The (F_x, F_y) points show how much longitudinal and lateral force is being exerted at a given time, where time is indicated by color. Plots on the right show the total magnitude of force exerted over time, with red envelopes showing the dynamic friction limit as influenced by weigh transfer.

The top plots of the front axle force show that braking torque was not solely applied for the purpose of slowing the vehicle; it plays another key role. As the vehicle brakes, weight shifts to the front axle where our trajectory uses that additional headroom to steer hard left and yaw the car! Notice how F_{yf} exceeds the static *friction circle* at this time. The rear axle, furthermore, is not insensitive to this loss of control authority. The bottom right plot shows how the rear axle saturates at $\approx 0.75\,sec$, using every available Newton of force for the remainder of the maneuver. The bottom left plot shows considerable oscillation in the final $200msec$ of the maneuver, and suggests Euler integration may be insufficient to track such high frequency dynamics at $dt = 9\,msec$.

V. Conclusion

In conclusion, this project provided an excellent opportunity to dive deep into the possibilities available in nonlinear control optimization. Several challenges however, lie ahead to take this maneuver from simulation to experiment. Foremost, a closed loop scheme must be developed around this open loop trajectory.

One option is to design a traditional controller to follow the optimal trajectory produced by our program. Such a controller could be formulated to use our array of inputs as a *feedforward* command, and to design first or second order *feedback* terms to counter state error.

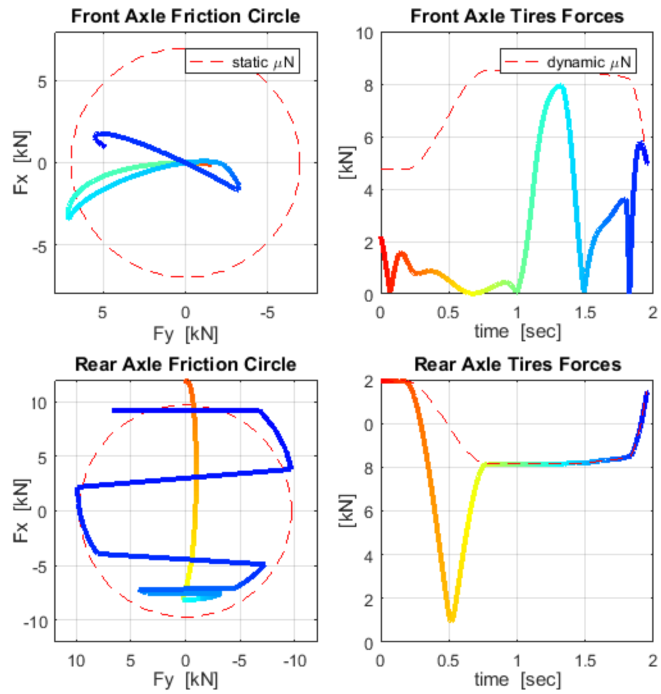

Fig. 6. 'Friction circle' (left) and magnitude (right) tire forces

A more ambitious idea is to deploy this nonlinear program as a *Model Predictive Controller*. In order to do so, solution times must reduced almost 3 orders of magnitude, from $\sim 100\,sec$ to maybe $100\,msec$. This tremendous leap may be accomplished in several ways: Implement the program in C++ as a compiled problem to eliminate the setup which currently occurs at run-time in MATLAB. IPOPT reports this setup currently occupies $\sim 90\%$ of our solve time. Second, a good initial guess may reduce computation time further. By feeding a previous solution back to IPOPT (not including dual variables), we observe that convergence occurs in about half the number of iterations. Another idea fruitful idea may be to linearize about the provided trajectory or relax the dynamics to a convex program.

Wish us luck! We're excited to press forward.

Acknowledgement

We would like to give our deepest appreciation to Mauro Salazar, Sumeet Singh, and Prof. Marco Pavone for their advice and guidance throughout this project.

References

[1] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 597–602.

[2] M. Acosta, s. Kanarachos, and M. Blundell, "Vehicle agile maneuvering: From rally drivers to a finite state machine approach," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, December 2016.

[3] E. Velenis, P. Tsiotras, and J. Lu, "Modeling aggressive maneuvers on loose surfaces: The cases of trail-braking and pendulum-turn," in *2007 European Control Conference (ECC)*, July 2007, pp. 1233–1240.

[4] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[5] A. Wchter and L. T. Biegler, “On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming,” in *Mathematical Programming*, vol. 106, no. 1, 2006, pp. 25–57.