Investigating the effects of roadside cover on safe speeds for autonomous driving in high-risk deer-vehicle collision areas

Joaquin Font, Alexander Brown

Department of Mechanical Engineering – Lafayette College,
740 High Street, Easton, PA, U.S.A. 18042
email: brownaa@lafayette.edu

Abstract

This paper explores the effects of roadside vegetation and/or cover on the probability of vehicular collisions with deer by identifying and simulating dangerous animal-vehicle interaction scenarios. Deer motion is described by a simplistic model inspired by the literature on deer escape behavior, and a genetic algorithm identifies problematic combinations of model parameter values given a particular vehicle-driver-environment configuration. A nonlinear, planar bicycle model is used to simulate vehicle dynamics. Two simulated driver-assist systems drive the vehicle model in simulation: the first applies brakes to approximate a "forward collision assist" system, and the second combines braking and steering inputs using model predictive control to avoid obstacles. Simulations of these driver-vehicle models interacting with deer are carried out for various configurations of roadside cover geometry. For the configurations explored in this study, the methodology produces recommended safe driving speeds for vehicles employing each driver assist system given a particular road configuration.

Keywords – Automated Vehicles, Simulation, Road Safety, Deer-Vehicle Collisions

1. Introduction

Estimates provided by the Insurance Institute for Highway Safety show that over 1.5 million deer-vehicle collisions occur per year in the United States, causing 150 occupant deaths, and over $1 billion in vehicle damage [1]. While a substantial body of literature addresses the danger of large animal collisions to humans [2], deer behavior and infrastructure-related factors in deer collisions [3, 4], patterns in deer-vehicle collisions [5], and human factors influencing the frequency of collisions [6], few studies have addressed how to accurately simulate intelligent vehicles interacting with animals on a roadway, and fewer still address collision avoidance system design with respect to animal-vehicle collisions specifically.

Considering the millions of miles of self-driving vehicle tests already being conducted on public roads [7], it is important to consider the safety of self-driving systems in the event of a sudden animal crossing. While there has been some recent work in the intelligent vehicles community focusing on animal detection, e.g. in [8, 9], more work is needed to prepare collision avoidance systems to encounter animals on the road, and/or to prepare for possible regulatory or infrastructure-related changes that may be necessary to make rural driving with self-driving vehicles as safe as possible. Combining what is known about animal-vehicle collisions, vehicular technology, and animal escape behavior will help inform Original Equipment Manufacturers (OEMs) and regulatory

bodies alike as level 3 and higher [10] vehicle automation penetrates vehicle markets, especially those markets that include rural driving. To this end, Cutrone et al. developed a simple, flexible model for deer motion during road crossing events [11]. A Genetic Algorithm (GA) was used to find particularly dangerous combinations of model parameters given a particular driver-vehicle system and sight distance to an animal's initial position. However, the paper stopped short of actually making recommendations for reducing collision risk. Further, the study only used rudimentary approximations of self-driving vehicles with very basic collision avoidance capability, and simulations did not incorporate the presence or proximity of roadside cover. The present study extends Cutrone's simulation methodology, incorporating more sophisticated driver models and including the effects of roadside cover on animal visibility to offer insights about how fast a particular vehicle-driver system should be permitted to travel on a particular road given its environmental geometry.

To develop a methodology for identifying deer-vehicle collision risk for a particular vehicle as a function of road cover geometry and speed, it was necessary to integrate knowledge from several different fields. Approaching the question of how fast a vehicle equipped with a particular collision avoidance system should go in high-risk areas for deer-vehicle collisions requires some understanding of how roadside cover affects collision probability (at least in human-driven vehicles), how deer may move when confronted with a probable collision, and what kinds of collision avoidance systems are likely to be employed in future self-driving cars.

Luckily, the effects of roadside cover on the frequency of deer-vehicle collisions have received a considerable amount of attention in the literature. For example, wooded areas are not only more likely to have higher deer density, but also hinder the driver and the sensors used in self-driving cars as they attempt to detect the presence of deer along the sides of the road [12]. In particular, Finder and Woolf [12] provided evidence that greater distance to forest cover decreased the probability of a road segment being a high deer-vehicle collision area. In other words, the closer the road was to a wooded area, the more likely a deer-vehicle collision was to occur. Considering collision data for Pennsylvania in particular, Bashore, Tzilkowski, and Bellis [13] found that high-frequency kill zones in wooded areas tended to be longer than in non-wooded areas, meaning that a driver could expect to encounter a deer almost anywhere in these areas. This paper also indicated that being able to see a deer early when trying to avoid a collision is important, and showed an inverse relationship between speed limit and collision probability. Further supporting the importance of sight distance in avoiding collisions with animals, Waring, Griffis, and Vaughn [3] studied deer crossings in both wooded and open field areas, and found that the lack of visibility in wooded areas was a key predictor of deer-vehicle collision frequency.

Compared with finding forensic data on deer-vehicle collisions, studies helpful for predicting how a deer or similar large animal might move when confronted with an imminent collision are more rare. Most of the work in this area is focused on how deer react when approached by predators. A vehicle may or may not be coded as a "predator" by a deer, but without a substantial body of data on deer motion when confronted with a car on a road in controlled, repeatable conditions, there is little else to use when building a dynamic model of animal flight. Additionally, while many of the studies describing deer motion in response to predator approach provide some metrics that quantify the gross behavior of the animals during flight, they generally do not provide a complete model of animal motion appropriate for simulation. For example, in [14], Lingle, Pellis, and Wilson studied escape gaits of different deer species in flight, but the study does not provide measures of flight direction. Some studies do provide some gross motion data on deer escape patterns, and some even mention how proximity to cover might influence behavior during flight. However, contradictions

between results presented in key studies such as the work of Rohm, Nielsen, and Woolf [15] and Grovenburg et al. [16] make sweeping generalizations difficult. If nothing else, this indicates that many complex factors likely interact to produce the trajectories that deer produce when fleeing. Significant factors influencing escape behavior aren't limited to landscape, either. Some researchers have even found that a deer's age can affect its escape behavior [17, 18]. For this reason, Cutrone's model of deer escape behavior, which was developed in [11] specifically to simulate deer flight during a road crossing event, used very general trends about the probabilistic nature of deer flight direction observed in [19] to prioritize model flexibility. The model allowed the simulated animal to change direction one time during a road crossing event. The model used in this study and presented in Section 2.1 is very similar to Cutrone's original model, allowing for wide ranges of model behavior to account for as much complexity in deer behavior as possible while keeping the number of model parameters to a minimum.

Finally, in considering interactions between deer and vehicles equipped with Advanced Driver Assist Systems (ADAS) or self-driving capability, it is important to use a representative model of a vehicle's collision avoidance system in simulation. While many approaches have been studied, algorithms that solve for an optimal input trajectory using a receding horizon approach have been relatively common for more than a decade, whether the optimization carried out is for path planning, or whether vehicle inputs are considered in the optimization directly [20]. For example, Falcone et al. [21, 22] have used Model Predictive Control (MPC) as a framework for vehicle control in multiple contexts. In [21], the authors developed an obstacle avoidance system capable of combined cornering and braking, and the optimization strategy directly produced vehicle control inputs that directed the vehicle around an obstacle. As is common [20], Falcone's controller in [21] also estimated the future position of an obstacle in order to allow the system to act in an anticipatory manner. This is key to avoiding obstacles that move, because the system must plan and execute a path that considers where the obstacle will be, instead of simply where the obstacle is now. While some approaches, including [21], solve for vehicle steering angle directly in the optimization framework, others, such as [23], plan a vehicle's path using a drastically simplified model. These approaches rely on low-level controllers to compute the required vehicle inputs needed to follow the optimal path.

While designing a controller that assumes a constant-velocity model of obstacle motion has been shown to produce good controller performance [21], more recent advances in obstacle avoidance technology have incorporated more complex probabilistic models of obstacle motion [24, 25]. Even though the MPC controller developed in Section 2.3 uses a constant velocity approximation for obstacle motion, the results presented in Section 3 could be replicated for any controller architecture, including those that have more sophisticated methods for predicting obstacle motion.

The remainder of this paper is organized as follows. Section 2 outlines the models for deer behavior, vehicle behavior, and driver behavior used in the experiments presented in Section 3. Section 4 discusses the results of two sets of simulation-based experiments that offer insights about maximum recommended speeds for the vehicle-driver model described in Section 2 given a particular roadside cover configuration, and Section 5 offers some general conclusions and recommendations for future work.

2. Simulation methodology

2.1. Deer motion model

In order to study the safety of an automated or partially automated driving system when confronted with a deer crossing the road, it is necessary to have a plausible model of how the animal's motion might evolve during the interaction. In [11], Cutrone et al. developed a rudimentary model of deer motion. The model represents a step up in complexity compared with a constant-velocity model, but remains flexible to accommodate a wide range of possible animal trajectories. In Cutrone's model, an animal's movement when crossing a road has the following three components or "phases."

1. Initial acceleration (phase 1)
2. Deceleration and turn (phase 2)
3. Final acceleration (phase 3)

In [11], a GA was used to explore a broad parameter space, eventually finding particularly dangerous combinations of parameters governing each phase of the simulated animal's motion given a particular driver model. Phase 1 represents the animal's initial decision to cross the road. Phase 2 represents a moment when the animal assesses that the risk of collision with the vehicle warrants a change in direction, and phase 3 represents the animal's final acceleration to a steady-state speed to complete the road crossing event. It is important to note that in implementing this approach, allowing a GA to search for dangerous parameter combinations should not be confused with an attempt to find *probable* deer crossing scenarios. The algorithm simply explores the space of approximate deer trajectories that are plausible under the assumptions that the deer participates in the "phases of motion" listed above while crossing the road. It is possible that the "most dangerous" deer crossing event as identified by the GA will never occur on a real roadway. However, when human fatalities are a risk, even low-probability events are important test cases for evaluating a vehicle safety system. As more studies explore deer flight behavior, the model structures and parameters of this framework can be easily updated to improve simulation fidelity.

The present study's implementation of the deer motion model preserves the overall three-phase structure of the model presented in [11], with limited changes to make the resulting motion more physical and improve the model's alignment with available data on ungulate flight behavior where possible. Figure 1 shows a plot of the forward velocity of the model through all three phases as a function of time, along with an illustration of key parameters as they progress through the three phases of motion.

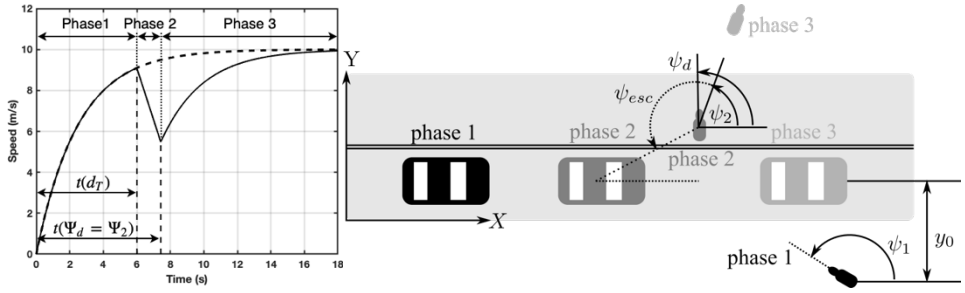


Figure 1. Deer model motion during a road crossing event

As Figure 1 shows, the simulated animal's speed follows a first-order growth towards a steady state value, which is interrupted by a period of constant deceleration. This deceleration occurs during the animal's direction change—the model assumes that large changes in angle require the animal to slow down to maintain a maximum "overall" acceleration. This total acceleration is governed by the deer's acceleration time constant. Once the turn to a final escape angle has been completed, the simulated animal accelerates back towards its steady state speed. This model for deer motion is fully described by only five adjustable parameters, which are shown with their respective ranges in Table 1.

Table 1: Deer model parameters and ranges

Symbol	Description	Range	Units
Ψ_1	Initial heading	[0,π]	rad
y_0	Initial offset from lane center	[-4.0,-30.0]	m
v_{max}	Steady-state forward velocity	[5.0,18.0]	m/s
τ_v	Acceleration time constant	[0.75,5.0]	s
d_t	Flight (phase 2) initiation distance	[20.0,120.0]	m

In phase 1, the model's velocity is governed by equation 1, approximating its velocity as first order.

$$\dot{v}_{deer} = \frac{1}{\tau_v}(v_{max} - v_{deer}) \tag{1}$$

In equation 1, v_{max} and τ_v were chosen to represent a range of plausible deer flight speeds and time constants, consistent with [11]. Global X (down the road) and Y (across the road) coordinates are updated in all phases according to equation 2.

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\end{bmatrix} = \begin{cases} v_{deer}\sin\psi_{deer} \\ v_{deer}\cos\psi_{deer}\end{cases} \tag{2}$$

In equation 2, Initial conditions are governed by the deer's initial offset from the lane center and longitudinal position. For consistency, and to match the distance at which the simulated vehicle controllers can detect an animal in the absence of occlusion (see Section 2.3), all simulated deer began moving at $x_0 = 60m$ farther down the road than the vehicle's initial position for all simulations in this paper.

As shown in Figure 1, ψ_{deer} is given by ψ_1 during phase 1. However, when the distance between the deer and the vehicle is less than or equal to the flight distance, d_t, the deer enters phase 2, marking the model's assessment that a collision is imminent. The range of distances allowed for d_t was taken from [19], in which the authors recorded flight behavior of black-tailed deer in response to foot pursuit by humans.

In phase 2, the deer selects a single flight direction ψ_{esc} from the normal distribution $\mathcal{N}(136°, (28.3°)^2)$. This distribution represents a fit of the data presented for deer flight directions in [19]. In the present model, ψ_{esc} is defined relative to the line connecting the deer and the vehicle at every instant, which means that the actual angle that the deer needs to achieve in the earth-fixed coordinate system depends on the vehicle's position.

Assuming that the deer's overall "urgency" in acceleration is governed by the single parameter τ_v, as in [11], the deer is permitted in simulation to accelerate in both the lateral and longitudinal direction with a maximum acceleration equal to the average acceleration that the deer experiences during the first τ_v seconds of phase 1. This maximum acceleration, a_{max}, is given by equation 3.

$$a_{max} = 0.632\frac{v_{max}}{\tau_v} \tag{3}$$

During phase 2, the simulated deer continuously computes the difference between its current orientation ψ_d and the angle through which it must turn to achieve its chosen escape angle ψ_{esc} relative to the car's position. In the earth-fixed coordinate system, this "goal angle" is given by ψ_2. At each timestep, $\psi_2$is given by equation 4.

$$\psi_2 = \tan^{-1}\frac{\mathrm{Y_{deer}}-\mathrm{Y_{car}}}{\mathrm{X_{deer}}-\mathrm{X_{car}}} + \psi_{\mathrm{esc}} \tag{4}$$

If turning to the goal angle in one timestep would result in a centripetal acceleration greater than a_{max}, the deer decelerates longitudinally with $\dot{v}_{deer} = -a_{max}$ and turns with maximum centripetal acceleration according to equation 3 during that timestep. This process repeats until either the car has passed, or the deer achieves the desired escape angle ψ_2. The maximum yaw rate the deer can achieve to maintain a lateral acceleration a_{max} is given by equation 5.

$$\dot{\psi}_{max} = \frac{a_{max}}{v_{deer}} \tag{5}$$

This per-timestep turning behavior results in smoother simulation trajectories than the model presented in [11], where the deer was assumed to make its full turn in a single simulation timestep. After the deer completes phase 2's deceleration and turn, it accelerates back towards its steady-state speed as given in equation 1 to complete phase 3. In all of the following simulations, the deer behavior described above is simulated using Euler integration in Python with a time step of 0.015s. For all experiments in Section 3, deer model parameters are "evolved" using a GA nearly identical to the one used in [11], with the GA attempting to minimize interquartile mean minimum distance between vehicle and deer during each test of a particular model configuration. In practice, this means that the GA attempts to find combinations of model parameters that are especially difficult to avoid. Just as in [11], The GA uses 5-bit resolution for each parameter, a population size of 15 genomes, and 8 trials per genome to compute a score for each. In cases where generations resulted in nearly identical scores across individuals, 5 randomly generated individuals were added to the mating pool to help prevent the algorithm from remaining stuck in a local minimum. While many modifications to this deer motion model are possible, any modifications resulting from improved knowledge of deer behavior or modifications that lend more conservatism to the results (e.g. allowing the deer to accelerate to top speed more quickly) could easily be substituted into the simulations described below.

2.2. Vehicle model

A planar, single-track, non-linear "bicycle model" was used to simulate the behavior of a car moving along the road given inputs for acceleration, braking, and steering. Pacejka's "magic formula" [26] was used to simulate the interaction between the tires and the road under combined

cornering and braking. The model is limited only to in-plane motion, and does not account for the effects of weight transfer, rolling, pitching, terrain disturbances, or changes in elevation. While this is a substantial limitation, a nonlinear bicycle model has been shown to capture much of the key vehicle behavior required for road safety simulations [27]. The vehicle state vector and coordinate system are illustrated in Figure 2 below.

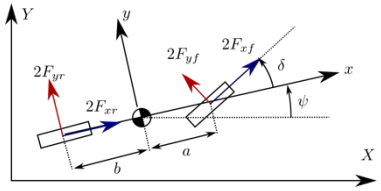


Figure 2: Vehicle model configuration and coordinate systems.

In Figure 2, the velocity of the vehicle in the local x-direction is given by U, and the velocity in the local y-direction is given by V. The equations of motion for the model are given in equation 6, which employs the small angle approximation for the roadwheel steering angle δ. During all experiments, vehicle dynamics were simulated using 4th order Runge-Kutta integration implemented in Python with a timestep of 0.015s.

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} Y \\ V \\ X \\ U \\ \psi \\ r \end{bmatrix} = \begin{cases} V\cos\psi + U\sin\psi \\ \frac{1}{m}\left[2F_{yf} + 2F_{xf}\delta + 2F_{yr}\right] - Ur \\ U\cos\psi - V\sin\psi \\ \frac{1}{m}\left[2F_{xf} - 2F_{yf}\delta + 2F_{xr}\right] + Vr \\ r \\ \frac{1}{J_{zz}}\left[a(2F_{yf} + 2F_{xf}\delta) - bF_{yr}\right] \end{cases} \tag{6}$$

In equation 6, tire forces $F_{yf}, F_{yr,}F_{x_f}$, and F_{xr} are computed based on Pacejka's formula for steady-state cornering behavior under combined cornering and braking [26], and are thus functions of both brake pressure and roadwheel angle δ. The roadwheel angle δ is computed assuming that the vehicle has a steering motor. The steering motor's behavior is approximated by a second order transfer function in standard form that relates desired steering angle δ_d to actual steering angle δ with a steady-state gain of 1 and ζ, ω_n of 0.9 and 15 rad/s, respectively. The vehicle mass and inertia properties were chosen to match a compact sedan with $m = 2000kg$, $J_{zz} = 1000kg \cdot m^2$, distance from CG to front axle $a = 0.8m$, and distance from CG to rear axle $b = 1.5m$.

2.3 Driver models

During all experiments described in Section 3, one of two simulated collision avoidance systems was used. The first, a simple "braking-only" controller, applies the brakes the moment an obstacle is visible, similar to a "forward collision assist" system in a commercially available vehicle. The other driver model was designed to approximate a collision avoidance system that might be present in a fully self-driving car. Both controllers react to the simulated deer as soon as it is visible, with visibility defined by whether the deer is "in the open" or occluded by roadside cover (vegetation). Specifically, the simulated controllers are able to detect and react to obstacles up to 60m away in

the absence of occlusion. This value could be adjusted for any configuration of sensors available on a vehicle. For both controllers, aggressive braking begins immediately once an obstacle is detected. In the case of the MPC controller described in Section 2.3.2, braking is modulated when the controller computes that steering is necessary to avoid a collision. Only single-obstacle scenarios are considered for all simulations presented in Section 3. To account for the effects of roadside cover on the success of a deer avoidance maneuver, a ray-casting algorithm was employed to detect whether the simulated deer was inside the boundaries of a roadside cover region. Each of the driver models is described below in more detail.

2.3.1 "Braking only" controller

This simplistic controller maintains a constant speed at the speed limit while searching for obstacles within its sensors' range. The moment an obstacle is detected, the algorithm applies the braking force required to achieve a longitudinal deceleration of 0.6g to the wheels, which is conservative but consistent with the possibility that the system may operate on wet roads [27]. This continues until a full stop is achieved. The utility of this controller is two-fold. First, it represents a rough approximation of commercially available forward collision assist systems, which only apply brakes. Second, it requires no complex computations in simulation other than ray-casting. This allows the GA to be iterated through many generations far more quickly than is possible with the MPC algorithm described in the next Section.

2.3.2 Model predictive controller

To approximate the behavior of a vehicle under SAE level 3 automation or higher as defined in [10], several of the animal-vehicle interactions simulated in Section 3 employ an MPC vehicle guidance and collision avoidance system. As mentioned in Section 1, MPC is a relatively common choice for vehicle guidance and path planning [20], because it involves the computation of an optimal input trajectory for the vehicle given an objective function and subject to constraints. A more thorough treatment of MPC, also called "receding horizon control," can be found in many texts, e.g. [28]. The controller used in this study was designed to balance tractability, computation time, and performance. While the goal here is certainly not to propose the best possible controller, a competitive, current algorithm was necessary to obtain simulations of interactions between animals and automated vehicles that capture key features of receding horizon control. It is important to note that any collision avoidance algorithm could be substituted into the simulations in Section 3 to obtain collision probability results for a particular vehicle.

Briefly, a discrete-time receding horizon controller predicts the state trajectory of a dynamic system N_p timesteps into the future for a given input trajectory. The controller then computes the value of some objective function J for the input vector $\vec{u}$, and uses an optimizer to calculate the optimal input trajectory $\vec{u}^*$ for which the value of J is minimized. The controller executes only the first input $\vec{u}^*(0)$ of the computed optimal input trajectory. Then, at the next timestep, the controller repeats the entire process to calculate the new optimal trajectory $\vec{u}^*$. This process creates implicit feedback, even though the optimal input trajectory $\vec{u}^*$ is itself an open-loop control policy. In the simulations presented in Section 3, the MPC predicts the vehicle's behavior and the obstacle's behavior simultaneously to allow the vehicle to react predictively to projected future obstacle positions.

The MPC system predicts $N_p = 10$ timesteps into the future with a prediction timestep $\Delta t_p = 0.1s$, which yields a temporal prediction horizon of 1 second for steering maneuvers. After

computing the optimal steering trajectory, the algorithm uses the first optimal steering input, $\vec{u}^*(0)$ to determine how hard the vehicle can brake without saturating the tires. It does this using a circular approximation of the tires' friction envelope according to Equation 7 below:

$$a_{x,max} = \sqrt{\gamma_b^2 - a_y^2} \tag{7}$$

In equation 7, accelerations a_x and a_y are lateral and longitudinal acceleration respectively. γ_b represents the radius of the friction envelope. All simulations of the MPC controller presented here used a conservative value of $\gamma_b = 0.6g$ to account for varying road friction conditions [27], lending conservatism to the controller's maneuvers.

All optimizations performed by the MPC were computed in Python using the Scipy library's SLSQP algorithm, which is an implementation of the algorithm in [29]. The specifics of the controller's predictions, objective function, and constraints are presented in the following subsections.

2.3.2.1. MPC predictive models

The inclusion of a complex vehicle model drastically increases the time required to compute an optimal input trajectory, and creates challenges in terms of objective function complexity and/or convexity. Therefore, the MPC developed here predicts the vehicle's behavior using a simplified kinematic model. This model takes braking/driving acceleration and roadwheel steer angle in radians as inputs, and assumes zero sideslip. While this is a considerable simplification, simulations of the controller indicate that it performs well, even under aggressive combined braking and cornering. This is likely due to the implicit feedback inherent in repeated optimization that is a hallmark of MPC, where predictive model errors are compensated for as the controller continues to react to the vehicle and obstacle's current states. Equations of motion for the kinematic model used in the present study is shown in equation 8.

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}_{kin} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} Y \\ V \\ X \\ U \\ \psi \\ r \end{bmatrix} = \begin{Bmatrix} U\sin\psi \\ 0 \\ U\cos\psi \\ \frac{F_{xf}+F_{xr}}{m} \\ r \\ \frac{1}{\tau_r}\left(\frac{U}{a+b}\delta - r\right) \end{Bmatrix} \tag{8}$$

In equation 8, the state variables and vehicle parameters (e.g. a, b) were matched to the nonlinear vehicle model presented in Section 2.2, and the time constant τ_r was chosen to approximate the yaw rate dynamics of the nonlinear vehicle model.

While the MPC's prediction of the vehicle dynamics is central to its ability to plan an input trajectory (and thus a path) for the vehicle over the prediction horizon, its prediction of the obstacle (deer) motion is equally important. Because suitable models of deer motion are scarce, and to avoid predicating the results presented in Section 3 on an advanced model for obstacle motion, the control system implemented in this study uses a constant velocity prediction of deer motion throughout the horizon.

2.3.2.2 MPC objective function and constraints

Much like the approach in [23], the objective function penalizes the vehicle's distance from the road at each timestep in the prediction horizon, along with its lateral velocity and acceleration. It also penalizes the commanded steering velocity to avoid unrealistic, rapid steering maneuvers. For all simulations in this study, the objective function J is given by equation 9.

$$J = \sum_{k=1}^{N_p} \left[q_1 \cdot \left(Y(k) - Y_{road}(k)\right)^2 + q_2 \cdot \left(\delta(k) - \delta(k-1)\right)^2 + q_3 \cdot \left(V(k)\right)^2 + q_4 \cdot \left(a_y(k)\right)^2 \right] \quad (9)$$

The weights in equation 9 were held constant for all experiments with [q1,q2,q3,q4] = [10,1,1,0.005], and were chosen heuristically to provide good performance when simulating interactions with constant-velocity obstacles.

In addition to minimizing the objective function, the controller must abide by a set of constraints which prevent the car from steering off the road and from hitting the obstacle at any point in the prediction horizon. This is similar to the structure of the constraints used for the receding horizon path planner in [23]. The set of constraints the MPC system is subjected to during each optimization is shown in equation 10.

$$\begin{cases} Y(k) - Y_{min}(k) > 0 \ \ \forall k = 1,2,\dots,N_p \\ Y_{max}(k) - Y(k) > 0 \ \ \forall k = 1,2,\dots,N_p \\ \sqrt{\left(Y(k) - Y_{obs,pred}(k)\right)^2 + \left(X(k) - X_{obs,pred}(k)\right)^2} - C > 0 \quad \forall k = 1,2,\dots,N_p \end{cases} \quad (10)$$

For simplicity, only straight roads were simulated for this study. Therefore, the road boundary constraints in equation 10 used constant values obtained from Google Earth measurements of a rural two-lane road in Bucks County, Pennsylvania, USA. For all simulations, the center of the lane was located at $Y_{road} = 0$, and lanes were 3.5m wide, with 2m wide shoulders. Consequently, Y_{min}, Y_{max} were set to -3.75m and 7.25m, respectively. Additionally, the value of C (the minimum distance to obstacle allowed in equation 10) was set to 2m for all experiments. In optimizing equation 9 given the constraints in equation 10, the MPC algorithm computed an optimal steering angle (and a subsequent allowable braking acceleration according to equation 7) every 0.1 simulated seconds. Each computation provided a requested roadwheel angle to the vehicle model's steering actuator. In between MPC computations, the desired steering angle was held constant.

To visualize how this MPC avoids a simple constant-velocity deer, a sample interaction with vehicle initial speed $U_0 = 25m/s$ is shown in Figure 4. For this interaction, the deer crossed the road with a constant velocity $v_{deer} = 13m/s$ at an angle of $\psi_1 = 173°$. The simulated constant-velocity deer began $x_0 = 60m$ ahead of the car and $y_0 = -2m$ from the lane center.

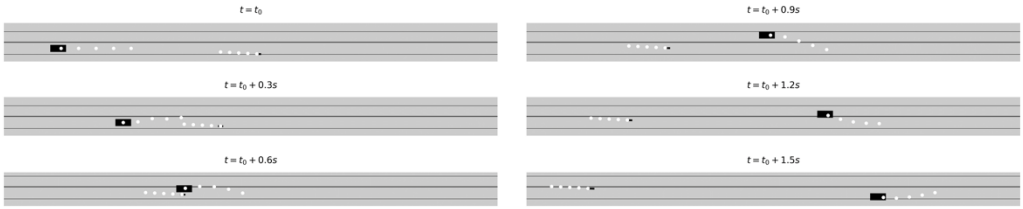


Figure 3: Predictions and vehicle/obstacle trajectories during example collision avoidance maneuver.

In the simulation presented in Figure 3, the vehicle decelerates long before it begins to steer, with the controller eventually producing an optimal steering and vehicle state trajectory over time that resembles a double lane change.

To validate the MPC system's performance, a battery of interactions between the vehicle-controller system described in Section 2.3.2 and a constant-velocity obstacle with speeds and angles spanning the ranges in Table 1 were simulated. The MPC was able to avoid collisions for 98% of the obstacle velocities (in 1 m/s increments) and angles (in 1 degree increments) simulated, indicating that it performs as intended in cases except those in which the obstacle traveled at or near maximum speed while oriented directly towards the vehicle.

3. Simulation Description

Two different series of simulations were conducted to explore the effects of roadside cover on deer-vehicle interactions. In both, roadside cover was assumed to completely occlude a simulated deer from the vehicle's sensors as described in Section 2.3. No other sources of animal occlusion or sensor failure were included. This choice keeps the simulation framework agnostic of vehicle sensor configuration. Obstacle detection methods vary, and not all are sensitive to the same environmental factors. For example, if a vehicle is equipped with LIDAR, it may be able to detect an animal just as well at night as in daylight. If a vehicle is only equipped with cameras for obstacle detection, lighting might become as important as physical occlusion in determining when an animal could be detected. If the simulation framework described here was applied to a specific vehicle with a specific sensor suite, these extra environmental factors could be included in the simulations as appropriate, but in the following simulations, physical occlusion is the only factor considered for generality.

The geometry of wooded areas on the side of the road was manipulated to explore effects of map geometry on allowable vehicle forward speed in the face of "worst case" deer model parameters, since wooded areas are known to increase the probability of collisions [13]. The first simulation experiment studied the effects of set-back distance for wooded areas that extend infinitely in front of and behind the car. The second simulation experiment implemented a procedure for automatically calculating maximum safe speeds given a rectangular patch of roadside cover. Both consisted of running simulations of an autonomous driver in different environments, and allowing the GA described in Section 2.1 to find challenging combinations of deer model parameters to assess collision risk. The two road configurations are shown in Figure 4.

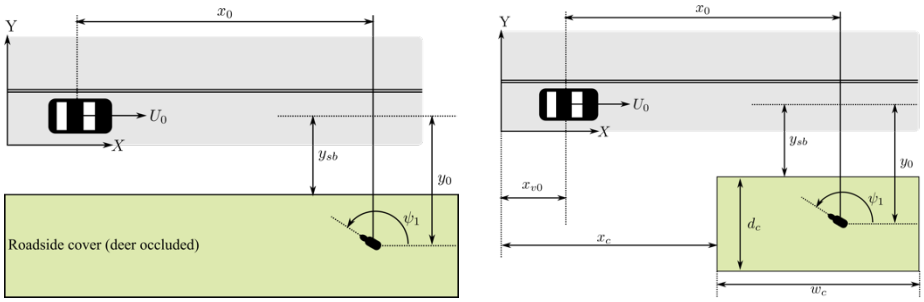


Figure 4. (a) Simulation experiment 1 road configuration (b) Simulation experiment 2 road configuration

3.1 Simulation experiment 1 description

In simulation experiment 1, four artificial maps were created to simulate the presence of a dense wooded area at a fixed constant distance from the lane center. The edge of the region of cover was assumed to be straight in the direction of the road, and to extend infinitely in front of and behind the vehicle. The simulated deer described in Section 2.1 started $x_{v0} = 60m$ ahead of the starting position of the vehicle in the X-direction for all simulations. Because the initial deer lateral position y_0 is controlled by the GA, the deer is able to "hide" inside the region of cover to allow it to move towards the vehicle without being detected. For vehicle speeds ranging from 25 m/s down to 15 m/s, the deer model was trained for 100 generations using the GA described in Section 2.1 by interacting with the "braking only" controller described in Section 2.3.1. The resulting "most difficult deer" for each speed was tested against both the "braking only" system and the MPC system for an additional 100 trials each to obtain a clear picture of the collision risk that the most difficult deer might pose to the vehicle/driver system.

This procedure was repeated for four different configurations of the simulated environment with cover setback distances of $4m, 7m, 10m$, and ∞m. Figure 4a shows the simulation's road configuration. With 12 speeds tested, 100 generations tested at each speed, 10 new genomes tested at each generation, and 8 deer-vehicle interaction simulations per genome, the results from simulation experiment 1 represent 8,000 individual deer-vehicle interactions for each setback distance. An additional 100 deer-vehicle interactions were used to provide a picture of the collision risk of the "worst-case" genome, as discussed in Section 4.1.

3.2 Simulation experiment 2 description

Although "infinite" wooded areas like those shown in Figure 4a may help provide general design or regulatory guidance when determining safe driving speeds for self-driving cars, a methodology that could incorporate actual maps of regions of cover to determine "map referenced" safe speeds would be even more valuable. Therefore, a second simulation experiment employed an algorithm designed to search for safe speeds as a function of vehicle position. The idea is that if these simulations were conducted off-line before a vehicle was driven on a specific road, the algorithm might be able to produce a maximum safe speed profile as a function of road station for the vehicle through a high-risk area. This has potential as both a vehicle design aid and as a framework for regulatory decision-making. In simulation experiment 2, the road and cover

geometry considered is a straight, flat, two-lane road with a single patch of roadside cover, as shown in Figure 4b.

For each initial vehicle location along the road x_{v0}, and with the deer beginning x_0 meters ahead of the car, deer-vehicle interactions were simulated with an initial vehicle speed U_0 corresponding to the road's speed limit. The GA attempted to find the "worst-case" deer genome by iterating through up to 20 generations. If the deer genomes produced in 20 generations did not result in any actual collisions (checked by approximating both vehicle and deer as rectangles), the vehicle moved down the roadway to check the next location in a similar way. If, however, one or more of the tests of a particular genome during training did result in a collision, the vehicle's initial speed U_0 was reduced, and the process was repeated until a "safe" speed for that road location was found. This algorithm is illustrated graphically in Figure 5.

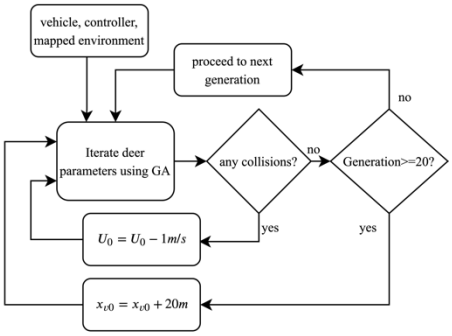


Figure 5: Map-referenced "safe speed" search algorithm

For this set of simulations, initial vehicle speed U_0 was chosen to be 25m/s (roughly 56mph) to reflect common speeds on rural highways in Pennsylvania. The deer's initial position was set to $x_0 = 60m$ ahead of the car for all initial vehicle station coordinates x_{v0} corresponding to the maximum sensor range without occlusion. This means that as the vehicle's initial position x_{v0} was modulated by the algorithm, so was the deer's initial position along the road. The deer's initial lateral position y_0 was controlled by the GA at each location and speed simulated.

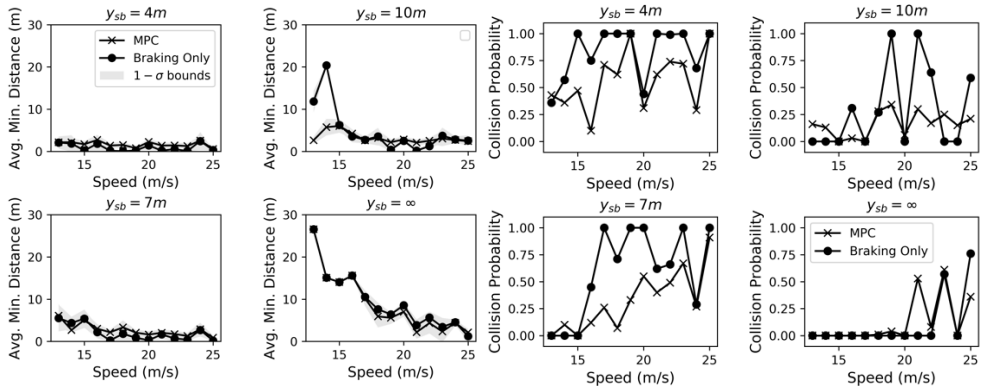
To simulate roads with a patch of wooded area, each environment was created with a single rectangular patch of trees. The patch used setback distances of $y_{sb} = 4m, 7m, 10m, \infty m$, a width $w_c = 80m$, and a depth $d_c = 90m$ as illustrated in Figure 4b. The wooded area began at $x_c = 80m$, so that varying x_{v0} between $0m$ and $120m$ in increments of $20m$ encompassed situations where the vehicle would encounter a simulated deer that began its motion before, within, and after the region of cover. In theory, any mapped road cover geometry and any vehicle configuration could be tested in the same way.

4. Results and Discussion

4.1 Simulation Experiment 1

Figure 6a shows the distribution of minimum distances as a function of speed for all four "inifinite cover" environment configurations, and Figure 7b shows collision probabilities for all

four environments and speeds, which were computed by modeling the deer and the vehicle as rectangles and checking for collisions for each of the 100 "test" simulations.



(a) (b)

Figure 6: Simulation Experiment 1 (a) average minimum distances, and (b) collision probabilities for varying speeds and cover setback distances

Figures 6a and 6b show that the MPC generally outperforms the braking-only driver in situations where the cover occludes the simulated deer. For the $y_{sb} = \infty m$ case, the controllers perform similarly, which makes sense because the vehicle is able to start braking at the full sensor sight distance of $60m$. This full visibility case does indicate that these controllers should likely avoid speeds over about 19 m/s in high-risk areas that do not have significant terrain effects limiting visibility. The simulations also indicate that collision probabilities are highest for the smallest value of $y_{sb} = 4m$, where even speeds down to 13 m/s could be problematic if the "worst case deer" identified by the GA after 100 generations did appear. This general trend, along with the trend observed in the minimum distances as functions of speed in Figure 4b, are in agreement with the general trends indicated in studies of deer-vehicle collisions [12,13]. However, this simulation methodology could be applied for any collision avoidance system deployed on any vehicle to get a quantitative idea of what speeds might be appropriate in wooded areas that can be approximated by the geometry in Figure 4a.

4.2 Simulation Experiment 2

For the road configuration shown in Figure 4a, the algorithm summarized in Section 3.2 and Figure 5 produced estimated "safe speeds" given the 20-generation threshold per speed. Table 2 shows the number of actual deer-vehicle interactions that produced the algorithm's speed output at each road position x_{v0} for each cover setback distance y_{sb}. Figure 8 shows the "safe speed profile" for each road configuration and the MPC-driven vehicle described in Section 2.3.

Table 2: Number of Deer-vehicle Interactions at each x_{v0} resulting from the safe speed search algorithm (simulation experiment 2)

x_{v0}	0m	20m	40m	60m	80m	100m	120m	140m	160m	180m
$y_{sb} = \infty m$	2,720	4,920	5,600	6,920	10,600	7,160	3,440	4,320	4,040	7,640
$y_{sb} = 10m$	4,968	3,712	7,240	6,680	6,200	7,160	3,000	3,200	3,280	1,880
$y_{sb} = 7m$	2,960	1,720	6,160	7,560	4,360	4,400	4,600	3,640	3,360	5,120
$y_{sb} = 4m$	4,720	6,040	4,920	5,960	4,160	7,160	3,680	2,960	3,240	2,680

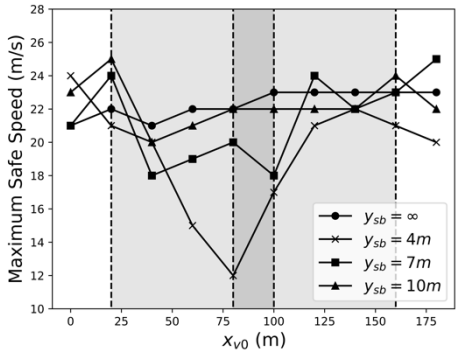


Figure 7. "Safe" speed profile resulting from simulation experiment 2

As the vehicle approaches the patch of roadside cover, Figure 7 indicates that it must slow down significantly to ensure that it will be able to avoid collisions with an animal that appears from behind cover, and that this effect is more pronounced as the patch of cover lies closer to the road's edge. This effect increases in severity with smaller road setback distances. It is also interesting to note that as the vehicle's initial position x_{v0} approaches 140 meters, the vehicle is still inside the region of cover but the simulated deer's initial position is not. Because the deer is assumed to begin moving $x_0 = 60m$ in front of the car for each set of simulations, the indicated "safe speed" increases. However, this should most likely not be taken as evidence that the vehicle should speed up while it is still driving through the region of cover. In reality, an animal could begin moving from behind cover at distances far smaller than 60m in front of the vehicle. Therefore, the results presented in simulation experiment 1 (Figure 6) may offer more conservatism for determining how fast a vehicle should travel while still within a forested region, because the GA's exploitation of the cover's ability to conceal a deer offers some approximation of a deer that begins its motion closer than 60m to the vehicle. Still, simulation experiment 2 (Figure 7) does offer quantitative measures of how a vehicle's speed should change as it approaches a region of cover. The algorithm predicts lower safe speeds with increasing setback distance between road and cover, and lower speeds while the vehicle is inside or approaching a region of cover.

Note that the results shown in Figure 7 indicate some variability in the identified safe low-risk speeds, even in the absence of cover. Anecdotally, we found that running this battery of simulations in repeated trials resulted in some variability between trials, even with consistent cover geometry. Some of this is likely due to the inherent stochasticity in the GA—new genomes are developed using randomly chosen combinations of operations on each parameter, so the GA may need more

than 20 generations to find a truly "worst case" parameter set. The 100-generation trials in simulation experiment 1 were far more repeatable. In general, choosing an appropriate number of generations is a balance between the thoroughness of the search of the parameter space and computation time.

Overall, these results show promise for finding speeds that will increase vehicle safety in areas where deer-vehicle interactions are likely. A vehicle manufacturer could substitute any control architecture into this simulation framework and obtain a safe speed profile for any road. This algorithm, run offline before a vehicle is released to market, could add a priori information to a vehicle's semantic map so that appropriate speeds can be selected in high-risk areas.

4.3 Limitations

While the two batteries of simulations presented here were designed to cover a range of possible road configurations and used a relatively modern approach to collision avoidance for the simulated self-driving vehicle, it is important to point out several important limitations of the study. Results in Section 4 are intended primarily to demonstrate the flexibility of the simulation framework as a tool for assessing collision risk, rather than to provide specific regulatory advice.

Both the vehicle and animal models in this study limit the generality of the results. The experiments only included one set of vehicle parameters and one obstacle avoidance controller, and it should be restated that higher-fidelity simulations of both vehicle and driver would improve simulation fidelity, perhaps at the cost of computation time. Additionally, the deer motion model developed in Section 2.1 is based on deer motion parameters from the literature, but many parameters use ranges that are estimated. All parameters used would benefit from increased availability of high-fidelity deer motion data.

The simulation configuration space explored here also carries some caveats when interpreting the results. Although the methods presented in Section 3 are easily adaptable to any road geometry, the geometries used in simulation experiments 1 and 2 are simplistic, and certainly do not cover the entire range of possibility for deer-vehicle interactions. As mentioned at the beginning of Section 3, sensor faults and variability in environmental conditions that could affect deer visibility are also not considered. These should be included if this framework is used for a particular vehicle with a known sensor suite. Finally, the assumption that the animal begins moving at a fixed distance (60m) in front of the vehicle is somewhat limiting, but is adjustable in the simulation framework.

The sensitivity of the results to many of these limitations is planned for investigation in a future study. However, the results presented in Sections 4.1 and 4.2 show that using a genetic algorithm to explore animal-vehicle collision risk as a function of road geometry and vehicle position can provide a quantitative picture of collision risk given a particular intelligent vehicle and a particular environment.

5. Conclusions and future work

This paper presented two simulation-based experiments that explored the effects of roadside cover on an automated collision avoidance system's ability to navigate around simulated deer. These deer exhibited stochastic behavior during interactions with a vehicle using a three phase model for their motion during an interaction, and the configuration of their trajectories was optimized to be as difficult to avoid as possible using a genetic algorithm.

In the first set of simulations, the deer behavior was iterated through 100 generations of the genetic algorithm for four different map configurations and 10 different speeds. The resulting high-

risk deer were tested for 100 trials against each vehicle controller. This procedure identified collision risk for each environment/vehicle/controller configuration as a function of forward vehicle speed. In the second set of simulations, an algorithm was iterated through vehicle start locations and speeds to automatically search for speeds that minimized collision risk as a function of vehicle position in a map.

The results of the two simulation experiments shown in Section 4 show the utility of the simulation framework. The results largely agree with intuition, indicating that reduced speeds for self-driving cars are less likely to result in deer-vehicle collisions, and that increased cover setback distances are desirable. By using semantic maps of roadside features that may occlude animals as they begin to cross a road, this methodology could provide OEMs with a path towards identifying safe target speeds for autonomous vehicles on rural roads in high-frequency deer-vehicle collision areas. The methodology presented here may also offer regulatory guidance when setting speed limits for rural self-driving vehicles, and the results do indicate that when a controller's modeled obstacle motion (constant velocity, in this case) and actual obstacle motion do not agree well, the control algorithm's performance may be compromised. Therefore, caution in selecting vehicle travel speeds is warranted in high-risk animal-vehicle collision zones.

Acknowledgements

The authors would like to acknowledge the helpful comments of Brent Utter and Chun Wai Liew throughout the project as members of the first author's undergraduate thesis committee.

References

1. "Facts Statistics: Deer vehicle collisions," III (2016). [Online] https://www.iii.org/fact-statistic/facts-statistics-deer-vehicle-collisions. [Accessed: 22-Sep-2018].
2. Jakobsson, L., Lindman, M., Carlsson, H., Axelson, A., Kling, A. (2015). Large Animal Crashes: the Significance and Challenges. *Ircobi*, 15(42), pp.302-314.
3. Waring, G.H., Griffis, J.L., Vaughn, M.E. (1991). White-tailed deer roadside behavior, wildlife warning reflectors, and highway mortality. *Applied Animal Behaviour Science*, 29(1-4), pp. 215-223.
4. Ujvári, M., Baagøe, H.J., Madsen, A.B. (2004). Effectiveness of acoustic road markings in reducing deer-vehicle collisions: a behavioural study. *Wildlife Biology*, 10(1), pp. 155–159.
5. Steiner, W, Leisch, F., Hackländer, K. (2014). A review on the temporal pattern of deer–vehicle accidents: Impact of seasonal, diurnal and lunar effects in cervids. *Accident Analysis & Prevention*, 66, pp. 168–181.
6. Mastro, L.L., Conover, M.R., Frey, S.N. (2010). Factors influencing a motorists ability to detect deer at night. *Landscape and Urban Planning*, 94(3-4), pp. 250–254.
7. Axios, "Uber's autonomous cars have driven 1 million miles," [Online]. https://www.axios.com/ubers-autonomous-cars-have-driven-1-million-miles-1513305489-f9a7e397-09c5-40f9-b3f0ca05da06cda7.html, [Accessed March 29, 2018].
8. Wagner R., Thom, M., Gabb, M., Limmer, M., Schweiger, R., Rothermel, A. (2013). Convolutional neural networks for night-time animal orientation estimation*. 2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 316–321.
9. Saleh K., Hossny, M., Nahavandi, S. (2016). Kangaroo Vehicle Collision Detection Using Deep Semantic Segmentation Convolutional Neural Network. *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-7.
10. SAE J3016-201401 (2014). Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. [Online] https://standards.sae.org/j3016_201401/f. [Accessed: 22-Sep-2018].

11. Cutrone, S., Liew, C.W., Utter, B., Brown, A. (2018). A Framework for Identifying and Simulating Worst-Case Animal-Vehicle Interactions. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1995-2000.
12. Finder, R.A., Roseberry, J.L., Woolf, A. (1999). Site and landscape conditions at white-tailed deer/vehicle collision locations in Illinois. *Landscape and Urban Planning*, 44, pp. 77-85.
13. Bashore, T.L., Tzilkowski, W.M., Bellis, E.D. (1985). Analysis of Deer-Vehicle Collision Sites in Pennsylvania. *The Journal of Wildlife Management*, 49(3), pp. 769-774.
14. Lingle, S., Pellis, S.M., Finbarr Wilson, W. (2005). Interspecific variation in anti-predator behavior leads to differential vulnerability of mule deer and white-tailed deer fawns early in life. *Journal of Animal Ecology*, 74, 1140-1149.
15. Rohm, J. H., Nielsen, C.K., Woolf, A. (2007). Survival of white-tailed deer fawns in southern Illinois. *Journal of Wildlife Management*, 71, pp. 851-860.
16. Grovenburg, T. W., Swanson, C., Jacques, C.N., Deperno, C.S., Klaver, R.W., Jenks, J.A. (2011). Female White-tailed Deer Survival Across Ecoregions in Minnesota and South Dakota. *The American Midland Naturalist*, 165(2), pp. 426-435.
17. Nelson, T.A., Woolf, A. (1987). Mortality of White-Tailed Deer Fawns in Southern Illinois. *Journal of Wildlife Management*, 51, pp. 326.
18. Schwede, G., Hendrichs, H., Wemmer, C. (1994). Early Mother-Young Relations in White-Tailed Deer. *Journal of Mammalogy*, 75, pp. 438.
19. Stankowich T., Coss, R.G. (2005) Effects of predator behavior and proximity on risk assessment by Columbian black-tailed deer. *Behavioral Ecology*, 17(2), pp. 246–254.
20. Paden, B., Čáp, M., Yong, S.Z., Yershov, D., Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1), pp. 33-55.
21. Falcone, P., Borrelli, F., Tseng, H.E., Asgari, J., Hrovat, D. (2008). A hierarchical Model Predictive Control framework for autonomous ground vehicles. 2008 *IEEE American Control Conference*, pp. 3719-3724.
22. Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3), pp. 566-580.
23. Ziegler, J., Bender, P., Dang, Z., Stiller, C. (2014). Trajectory planning for Bertha—A local, continuous method. *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 450-457.
24. Karasev, V., Ayvaci, A., Heisele, B., Soatto, S. (2016). Intent-aware long-term prediction of pedestrian motion. *2016 IEEE International Conference on Robotics and Automation (ICRA),* pp. 2543-2549.
25. Wan, N., Zhang, C., Vahidi, A. (2017). Probabilistic Anticipation and Control in Autonomous Car Following. *IEEE Transactions on Control Systems Technology*, pp. 1–9.
26. Pacejka H. (2005). Tire and vehicle dynamics. Elsevier.
27. Brown, A., Brennan, S. (2014). On the required complexity of vehicle dynamic models for use in simulation-based highway design. *Journal of safety research*, 49, pp. 105-e1.
28. Kwon, W., Han, S. (2005). Receding horizon control. Advanced Textbooks in Control and Signal Processing. Springer.
29. Kraft, D., (1988). A software package for sequential quadratic programming. Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt.