Mathematical Modelling of Vehicle Trajectories Using the Frenet Coordinate System

Reece Palmer

I. Introduction

In differential geometry, the Frenet–Serret formulas describe the kinematic properties of a particle moving along a differentiable curve in three-dimensional Euclidean space $\mathbb{R}^3$, or the geometric properties of the curve itself, independent of motion. Specifically, they detail how the derivatives of the tangent, normal, and binormal unit vectors are expressed in terms of one another. Similarly, the simplest action describing the motion of a particle is proportional to the proper time along its trajectory in spacetime, or worldline. In the absence of external fields, this action is the unique Poincaré invariant action that can be constructed using only the velocity of the particle, ensuring consistency between kinematic and dynamic descriptions in a relativistic framework (Arreaga et al., 2001).

The Cartesian coordinate system is commonly used to determine the position of a vehicle. It is a fixed, global reference frame where the position is defined by a set of coordinates (x, y, z) along perpendicular axes, independent of any object's motion (Yagan, 2006). However, the Frenet coordinate system is more advantageous. Frenet coordinates are commonly used in trajectory tracking as they simplify the control problem by converting global Cartesian coordinates into a local reference frame aligned with the trajectory. This allows for easier separation of lateral and longitudinal control, enabling independent optimisation of each aspect of the vehicle's motion (Jiang et al., 2024).

The Frenet frame is a local coordinate system that adapts to the vehicle's path (Reiter et al., 2022), defined by three mutually perpendicular vectors: the tangent vector $\mathbf{T}$, the normal vector $\mathbf{N}$, and the binormal vector $\mathbf{B}$, which are determined dynamically based on the vehicle's instantaneous position on the road. The Frenet–Serret formulas governing the relationship between these vectors are:

$$\frac{d\mathbf{T}}{ds} = \kappa\mathbf{N}, \quad \frac{d\mathbf{N}}{ds} = -\kappa\mathbf{T} + \tau\mathbf{B}, \quad \frac{d\mathbf{B}}{ds} = -\tau\mathbf{N},$$

where s is the arc length, κ is the curvature, and τ is the torsion. Unlike the Cartesian system, the Frenet frame rotates with the vehicle's trajectory, making it inherently more suited for modelling and controlling the motion of self-driving vehicles, as it provides a relative and path-following framework, in contrast to the absolute Cartesian system, which is less intuitive for path-dependent motion.

To describe the motion of a vehicle or particle in terms of its kinematic properties, we define the variables ξ and d, representing the longitudinal arc length along the path and the lateral displacement from the path, respectively. For both variables, we identify their position, velocity, and acceleration, which are denoted as follows:

$\xi_i, \dot{\xi}_i, \ddot{\xi}_i$	initial position, velocity, and acceleration (along ξ),
$\xi_f, \dot{\xi}_f, \ddot{\xi}_f$	final position, velocity, and acceleration (along ξ),
$d_i, \dot{d}_i, \ddot{d}_i$	initial position, velocity, and acceleration (along d),
$d_f, \dot{d}_f, \ddot{d}_f$	final position, velocity, and acceleration (along d).

Since no explicit relationship between ξ and d has yet been established, they are treated as independent variables. The kinematic evolution of the vehicle can then be described using these variables and their respective properties. For example, the position of the vehicle in the Frenet frame is represented as (ξ, d), where ξ follows the arc of the path and d describes the lateral deviation from it.

We can formalise the motion planning problem as follows: Given a start configuration q_{start} (obtained from localisation and sensors), a goal configuration q_{goal}, and environmental constraints (such as physics, map data, and traffic), the goal is to determine a sequence of moves in the configuration space of the vehicle. The configuration space encompasses all possible states of the vehicle, often represented as $[x, y, \theta]$, where θ is the vehicle's heading. This sequence of moves must guide the vehicle from q_{start} to q_{goal} while avoiding obstacles and adhering to the constraints of the environment.

The overall path planning process can be decomposed into three primary subsystems: prediction, behaviour planning, and path planning. Prediction focuses on understanding and forecasting the actions of other vehicles over a given time horizon, enabling the self-driving vehicle to avoid collisions while optimising its path.

A. *Prediction*

To predict the actions of other vehicles, we can handle multi-modal uncertainty by modelling the prior beliefs about an object's motion using a Gaussian distribution. This allows the system to account for variability and uncertainty in the motion of surrounding vehicles, making it possible to predict where a vehicle is likely to move. The prediction pipeline typically employs two main approaches: a model-based approach and a data-driven approach.

The model-based approach incorporates knowledge of physics, road constraints, and vehicle dynamics, making it computationally efficient and suitable for real-time applications. Conversely, the data-driven approach leverages machine learning to identify patterns that may not be captured by deterministic models. One example of a data-driven approach is trajectory clustering. By analysing input data of past polynomial trajectories taken by other vehicles, an unsupervised machine learning algorithm can define a set of representative base paths. The probability of a particular action can then be determined by fitting the observed trajectory of a vehicle to these base paths.

A Gaussian Naive Bayes classifier is an effective method for integrating these approaches to handle uncertainty and improve predictions. The prediction process consists of two key steps:

1. Compute conditional probabilities: For a given data point, such as an $[x, y]$ pairing, the algorithm calculates the conditional probabilities using Gaussian distributions. These distributions are defined as:

$$p(x = v \mid C) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(v-\mu)^2}{2\sigma^2}\right),$$

where v is the observed value, C represents the class label, and μ and σ are the mean and standard deviation of the feature x for class C, computed during the training process.

2. Select the optimal hypothesis: The Naive Bayes classifier integrates the conditional probabilities from step 1 to determine the most likely class C by maximising the posterior probability. This is achieved using Bayes' theorem, under the assumption that features are conditionally independent:

$$\hat{C} = \arg\max_{C} \, p(C) \prod_{i} p(x_i \mid C),$$

where $p(C)$ represents the prior probability of class C, and $p(x_i \mid C)$ is the likelihood of feature x_i given C, calculated using Gaussian distributions. Research has shown that this probabilistic framework is particularly effective in scenarios with small datasets and clearly defined statistical distributions, ensuring robust predictions even with minimal training data. To extend this principle into behaviour planning for autonomous driving, the problem can be framed as creating a collision-free, smooth, and efficient path from start to destination based on inputs like the map, route, and predictions of other objects.

B. Behaviour Planning

Behaviour planning comprises generating a safe, efficient, and collision-free path from the start configuration q_{start} to the goal configuration q_{goal}, considering the map, route, and predictions of other objects. The objective is to create a trajectory that satisfies the conditions:

$$\text{Feasible, Safe, Legal, Efficient.}$$

A Finite State Machine (FSM) is used to structure the decision-making process. Each state represents a driving manoeuvre, and transitions between states depend on the vehicle's environment and conditions.

FSM

- **States:**
$$S_1 : \text{Decelerate to Stop}, \quad S_2 : \text{Track Speed}.$$
- **Transition Conditions:**
$$S_1 \to S_2 : \text{Distance to Stop Sign} \geq \text{Threshold},$$
$$S_2 \to S_1 : \text{Distance to Stop Sign} < \text{Threshold}.$$
- **Entry Actions:**
$$S_1 : \text{Set StopPoint = StopLine},$$
$$S_2 : \text{Set SpeedLimit = 50 \quad km/h}.$$
- **Transition Function:**
$$T(S_1, \text{Distance} \geq \text{Threshold}) = S_2,$$
$$T(S_2, \text{Distance} < \text{Threshold}) = S_1.$$

C. Path Planning

Path planning that moves the vehicle while adjusting its speed appropriately. The trajectory must be time-dependent, so we represent it with a 3-dimensional vector $[s, d, t]$, where t is time. The path must account for the positions of other vehicles at each time step, making driving in traffic a 3-dimensional problem. To create a successful trajectory, it must satisfy specific requirements. A key condition is minimising jerk, as higher jerk values lead to uncomfortable driving. By focusing on minimising jerk, we can model a smoother trajectory.

The total squared jerk is defined as:

$$J = \int_0^T \left(\frac{d^3 s(t)}{dt^3}\right)^2 dt$$

where $s(t)$ is the position along the path, and the third derivative of $s(t)$ represents jerk. Our goal is to minimise J by finding an optimal $s(t)$.

To minimise jerk, we note that all time derivatives of $s(t)$ beyond the sixth order must be zero. This leads us to define $s(t)$ as a polynomial with six coefficients:

$$s(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5$$

Similarly, the lateral displacement $d(t)$ follows the same process.

Next, we compute the velocity and acceleration by taking the first and second derivatives of $s(t)$, respectively:

$$v(t) = \frac{d}{dt} s(t)$$
$$a(t) = \frac{d^2}{dt^2} s(t)$$

When we set the initial time $t = 0$, we find that:

$$s(0) = 0, \quad v(0) = 0, \quad a(0) = 0$$

This simplifies the equations, and we can represent the system in matrix form:

$$\mathbf{A} \cdot \alpha = \mathbf{b}$$

Where $\mathbf{A}$ is the matrix of time coefficients, α is the vector of unknown coefficients, and $\mathbf{b}$ is the output vector.

To solve for the missing coefficients $\alpha_3, \alpha_4, \alpha_5$, we take the inverse of the matrix $\mathbf{A}$ and multiply it by the output vector $\mathbf{b}$:

$$\alpha = \mathbf{A}^{-1} \cdot \mathbf{b}$$

This gives us the coefficients, and we can substitute them into the original polynomial equation for $s(t)$. The resulting polynomial provides a viable path for the vehicle, ensuring smooth and efficient motion.

II. Discussion

Throughout this paper, we discuss the Frenet Coordinate System relative to prediction, behaviour planning, and path planning for vehicle trajectories. We find that the Frenet–Serret formulas describe the relationships between tangent, normal, and binormal vectors along a curve, with curvature (κ) and torsion (τ) governing their evolution. In the context of prediction, Gaussian-based methods and Naive Bayes classifiers are employed to probabilistically predict the motion of other vehicles, accounting for uncertainty in their trajectories. For path planning, we use polynomial models to minimise jerk, ensuring smoother and more comfortable vehicle trajectories by optimising the acceleration profile.

Additionally, we explored the fundamental concepts behind these techniques and demonstrated how they can be applied to improve the accuracy and efficiency of trajectory prediction and planning. However, the interpretations presented here serve as a foundation, and further research is needed to refine the mathematical models. Specifically, future work should focus on enhancing the Frenet–Serret framework by incorporating higher-order curvature and torsion terms to model more complex paths, including paths with varying torsion and non-constant curvature. Following, we advise further research prospects.

A. Interrelations Between Variables and Novel Modelling Concepts

Curvature (κ) and Lateral Displacement (d)

We treat ξ (longitudinal arc length) and d (lateral displacement) as independent variables. However, lateral displacement d can be influenced by the curvature κ of the road.

Proposed Idea: A model where $d(t)$ depends on curvature $\kappa(t)$, lateral velocity $\dot{d}(t)$, and lateral acceleration $\ddot{d}(t)$, such as:

$$d(t) = f(\kappa(t), \dot{d}(t), \ddot{d}(t)).$$

Jerk Minimisation and Frenet Frame

Jerk minimisation ensures smoother trajectories. The Frenet–Serret formulas describe the evolution of tangent (T), normal (N), and binormal (B) vectors, which are governed by curvature κ and torsion τ.

Proposed Idea: Incorporate jerk minimisation into curvature and torsion calculations. For instance:

$$\min J = \int_0^T \left(\frac{\mathrm{d}^3 s(t)}{\mathrm{d}t^3} \right)^2 \mathrm{d}t,$$

where $s(t)$ is the longitudinal position, with:

$$\frac{\mathrm{d}^3 s(t)}{\mathrm{d}t^3} = g(\kappa(t), \tau(t)).$$

B. Interaction Between ξ and d

We assume ξ and d are independent. However, lateral displacement d may affect longitudinal acceleration $\ddot{\xi}(t)$, for instance, during sharp turns or lane changes.

Proposed Idea: Add a coupling term:

$$\ddot{\xi}(t) = h(d(t), \dot{d}(t), \kappa(t)).$$

Higher-Order Derivatives

We focus on minimising jerk, but higher-order derivatives (e.g., snap or crackle) could further refine trajectory smoothness.

Proposed Idea: Extend the polynomial for $s(t)$:

$$s(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5 + \alpha_6 t^6,$$

and include higher-order terms in the optimisation process for improved comfort and precision.

REFERENCES

Arreaga, G., Capovilla, R., & Guven, J. (2001). Frenet–serret dynamics. *Classical and Quantum Gravity*, *18*(23), 5065–5083. https://doi.org/10.1088/0264-9381/18/23/304

Jiang, T., Liu, L., Jiang, J., Zheng, T., Jin, Y., & Xu, K. (2024). Trajectory tracking using frenet coordinates with deep deterministic policy gradient. *ArXiv*. https://www.arxiv.org/abs/2411.13885

Reiter, R., Nurkanović, A., Frey, J., & Diehl, M. (2022). Frenet-cartesian model representations for automotive obstacle avoidance within nonlinear mpc. *ArXiv*. https://arxiv.org/abs/2212.13115

Yagan, M. F. (2006). Coordinate system and coordinate transformations based on wave nature of light. *ArXiv*. https://doi.org/10.48550/arxiv.physics/0606036