# Module 4: Dynamic Object Interactions

## Lesson 1: Motion Prediction

内容
- Define the motion prediction problem for dynamic objects.
- Grasp the requirements for motion prediction.
- Understand the assumption for simplifications. Vehicles, Pedestrians.
- Constant Velocity Prediction Model. Issues.

Motion Prediction - Definition
- Motion prediction of the dynamic object's attempts to estimate the future position, heading and velocity.
- Important as it allows:
  - Plan a set of maneuvers to correctly interact with dynamic objects.
  - Avoid collisions on a planned trajectory.

Requirements for Motion Prediction Models
- Mandatory Requirements.
  - Class of Dynamic Object.
    - 理由： most prediction models have different algorithmic approaches to vehicles as opposed to pedestrians.
  - Current position, heading and velocity.
- Optional Requirements.（大事）
  - History of the position, heading and velocity.
    - Requires object tracking between identifications over a set amount of time.
  - Current high definition roadmap.
  - Image of the current dynamic object.
    - For vehicles, the image can provide information concerning the current indicator light or brake light states, for example.
    - For pedestrians, the image can serve to show the current orientation of the person, which can help predict a future direction of travel, even if the pedestrian is currently stationary.

Simplification of Motion Prediction - Cars
- Physics-based Assumptions.
- Maneuver-based Assumptions. 例：
  - They will most likely stay in their lane unless indicating otherwise and and stop at regulatory elements requiring stops.
  - They are unlikely to drive over sidewalks or lawns or through obstacles.
- Interactions-aware Assumptions.（大事）
  - The dynamic objects will react and interact with each other.
  - An example of this type of prediction, is during a merge by a vehicle into an adjacent lane.
  - Often, the vehicle in the destination lane will slow down to make more room for the incoming vehicle to maintain a safe following distance.

Complexities of Motion Prediction - Pedestrians.
- Physics-based.
  - Pedestrians tend to have a low top speed but can change their direction of motion and speed very quickly.
  - The range of positions a pedestrian can reach in a short time frame is limited.
- Maneuver-based.
  - Pedestrians tend not to interact directly with vehicles.
- Pedestrians ultimately have the right of way and it is the self-driving cars duty to stop when necessary.

  - どんな状況でも人の優先度は車より高い！
  - Inattentive pedestrians may wander into a roadway without warning, but will often stop when threatened by an oncoming vehicle.

Constant Velocity Prediction Model
- Simple. Computationally efficient.
- Assumption is that the dynamic object will maintain its velocity. Magnitude, Heading.
  - もし履歴なし、current状態のみ知っていれば、このモデルしかできないでしょう。

Constant Velocity Prediction Model - Issues
- straight line segmentにのみOK、Where the constant velocity estimate fails, however, is everywhere else.
  - weakly falls into the category of physics-based assumptions.
- Don’t account for Vehicle Dynamics fully.
- Don’t account for the Road (Position adjustment).
- Don’t account for Road Signs (Velocity adjustment).
  - Vehicles approaching stop signs tend to slow down and vehicles leaving a stop line tend to accelerate.
- Assumptions are too Strong and Incorrect for most Dynamic Object Motion.

単語
- Indicate: When drivers indicate, they make lights flash on one side of their vehicle to show that they are going to turn in that direction.
- Warning: A warning is an advance notice of something that will happen, often something unpleasant or dangerous.
- Horizon: Your horizons are the limits of what you want to do or of what you are interested or involved in.

# Lesson 2: Map-Aware Motion Prediction

内容
- A set of assumptions made by map-aware algorithms to improve motion prediction.
- Define a lane follow method to improve positional prediction.
  - Identify strategies to handle multiple future lane choices.
- Methods for velocity modulation around regulatory elements.
- Issues and short-falls with the map-aware assumptions.

Assumptions to Improve Prediction
- Positional Assumptions.
  - Vehicles on driving lane usually follow the given drive lane.
  - Changing drive lanes is usually prompted by an indicator signal.
    - なぜpositional assumptionという名前？ This is possible if the other vehicle is in a position to perform a lane change in the map.
- Velocity Assumptions.
  - Vehicles usually modify the velocity when approaching restrictive geometry (tight turns).
  - Vehicles usually modify the velocity when approaching regulatory elements.
- Important to understand that the more constraints or assumptions that are added to a prediction model, the less generalizable it can be to all traffic scenarios.
- In fact, there are cases where even these generic assumptions can be overly constraining.

Improvement of Position Estimation
- The predicted path is set to follow the center of the driving lane which the dynamic vehicle is on.
- Problems with the model.
  - Difficult to predict lane change maneuvers without extra information.
  - Multiple possible lanelets such as when on an intersection.
- Solution with the model.
  - Most likely prediction.

- Multi-hypothesis prediction. 車両予測のやり方だ。
    - Construct a prediction for the most likely behaviors and associate a probability that the agents will follow a particular path based on the state, appearance and track information.

Multi-hypothesis Approach
- Consider the range of all possible motions.
    - Left, right, stay stopped.
    - Based on corroborating evidence such as indicator signals, position to the left or right of the center line, and the state, of the vehicle at the intersection.
    - These probabilities can be learned from training data of many vehicles proceeding through similar intersections, or can be engineered and refined from real-world testing. 車両予測のやり方だ。
- Such approaches traditionally provide more ambiguous information to the behavior planner.
- Safer due to human error (forgotten turn signal).

Improvements to Velocity Prediction
- Road curvature can be used to improve the velocity prediction over the path.
    - Maximum lateral acceleration: $0.5 \sim 1m/s^2$.
- Improve the velocity prediction based on regulatory elements in the environment.
    - Stop locations, deceleration profiles.
    - Lanelet priors.
        - Given a HD road map, it is possible to preprocess the map for nominal trajectories along each roadway, and to define lanelet specific multi-hypothesis priors, based on nominal driving behavior.
        - This serves both as guidance for the ego vehicle in planning its behaviors and trajectories, and also in terms of refining the motion predictions for other agents.

Issues with the Assumptions
- Vehicles don’t always stay within their lane or stop at regulatory elements.
- Vehicles off of the road map cannot be predicted using this method.
    - such as a pothole in the road ahead or a bouncing ball.
- The best approach is therefore to track the evolution of beliefs over the set of hypotheses, and to update based on evidence from the perception stack at every time step.
    - 紹介なし?

## Lesson 3: Time to Collision

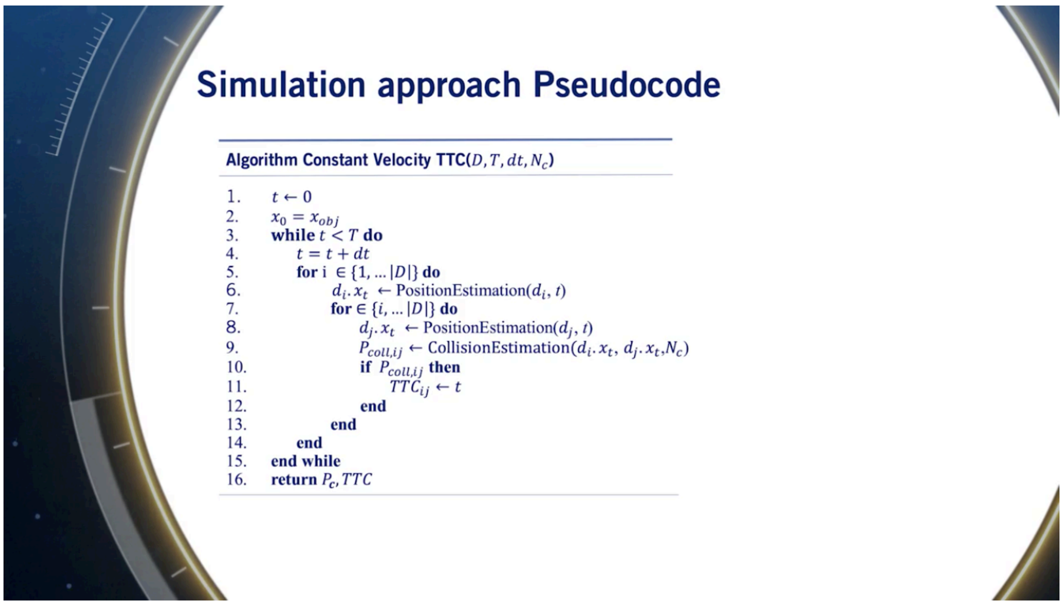
内容
- The concept of time to collision.
- Two approaches to calculate time-to-collisions and discuss their strengths and weaknesses.
    - Simulation approach.
    - Estimation approach.
- Outline a simulation based algorithm to calculate time-to-collision.

Definition of Time to Collision
- Assuming all dynamic objects continue along their predicted paths.
    - Will there be a collision between any of the objects?
    - If so how far into the future?
- Time to Collision is comprised of.
    - Identify and compute the location of a collision point along the predicted paths between the dynamic objects.
    - Prediction of the time to arrive at the collision point.
- Requirements for Accuracy.
    - Accurate predicted trajectories for all dynamic objects (position, heading and velocity).
    - Accurate dynamic objects geometries.
- Due to errors in both requirements, the time-to-collision should always be treated as an approximation, updated regularly, and treated with some safety buffer when making decisions about driving.

Simulation Approach
- Simulate the movement of each vehicle as time passes.
- Checking if any part of the two dynamic object has collided.
  - ADSTのやり方だ。
  - しかし実車でReal-timeでSimulationを行うの? いいえ、下記のRelative Strengths and Weaknessesを参考。確かにReal-timeでCARLAなどを動かすのが無理だ。
  - しかし、offlineのtime-to-collisionデータってどういう意味? 使えるの? アルゴへのサポート証拠としても意味あるけど。
- Note that this is a different type of collision checking than in the static case, where swaths of the entire path were constructed and compared to static object locations（Occupancy Grid Mapのやり方）.
- Because these collision checks are performed between moving objects, we only need to check collision between the geometries of each object at each instant, as the objects will occupy new locations at the next time step.

Estimation Approach
- Geometries of the vehicles are approximated over duration of the predicted path.
  - The results is a swath for each vehicle that can then be compared for potential collisions.
  - しかしswathが交差しても、タイミングがズレれば、問題にならないでしょう。タイミングはきっと考えるよ。
- Once the swath has been computed, their intersection can be explored for potential collision points by computing, if any two or more dynamic objects will occupy the same space at the same time.
  - Simulation based方法とどう違うんだ?
- Simplifying Assumptions.ちょっとやばい。
  - identifying collision point locations based on object path intersection points.
  - estimating object space occupancy based on simple geometric primitives like bounding boxes.
  - estimating the time to a collision point based on a constant velocity profile.
  - 簡単にいうと、これはまさにExcel-based計算だ。
  - Simulation-based方法はもちろんMotion Model、HD Mapなど必要ですが、車両運動の同期はやってくれるので、衝突判定は確かに楽かも。Simulationには条件、制限、策いくつあっても入れ込めるので、便利でしょう。

Relative Strengths and Weaknesses



## Simulation approach Pseudocode

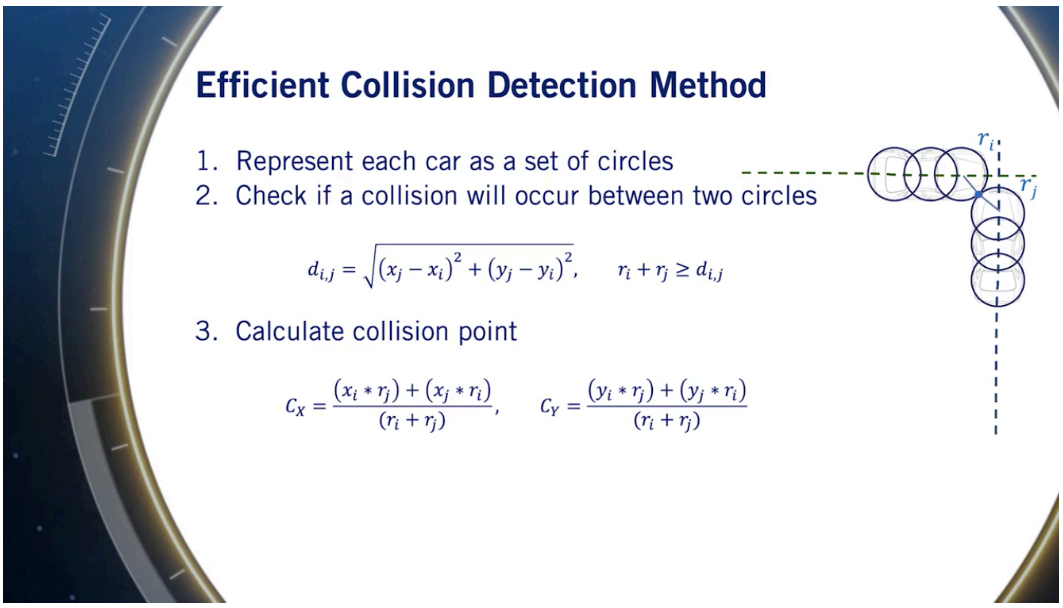
**Algorithm Constant Velocity TTC**($D, T, dt, N_c$)

```
t ← 0
x_0 = x_obj
while t < T do
    t = t + dt
    for i ∈ {1, ... |D|} do
        d_i.x_t ← PositionEstimation(d_i, t)
        for ∈ {i, ... |D|} do
            d_j.x_t ← PositionEstimation(d_j, t)
            P_coll,ij ← CollisionEstimation(d_i.x_t, d_j.x_t, N_c)
            if P_coll,ij then
                TTC_ij ← t
            end
        end
    end
end while
return P_c, TTC
```

| **Simulation Approach** | **Estimation Approach** |
|---|---|
| Computationally expensive. | Computationally inexpensive. |
| Higher accuracy if simulated with high fidelity. | Less accurate due to approximations and estimations. |
| Offline Applications (Dataset evaluation or Simulations).<br>- dataset creation.<br>- simulation-based testing. | Real Time Applications (Car Prediction).<br>- onboard time to collision computation. |

Simulation approach Pseudocode
- Inputs.
  - $D$: list of all dynamic objects.
    - with their predicted paths.
    - including the planned path for ego vehicle.
  - $dt$: time between simulation steps.
  - $N_c$: number of circles for collision approximation.
    - circle checking.
    - and their spacing relative to the object position.
    - 今ADSTの衝突判定はpolygon intersectionでしょう。
- Outputs.
  - $P_{coll}$: list of all collision points.
  - $TTC$: list of all times to collision points.

Estimation of Dynamic Object State
- For state prediction, we must take into account the possibility that the predicted path state time steps, do not align with the simulation time steps.
- Each predicted vehicle state has a predicted time at each location.
- Find the closest vehicle state along the predicted path to the current simulation time.
  - 内挿で同じtimeにしない? している!
  - We then apply the predicted inputs for the remaining interval of time to arrive at a prediction of the object location, at the current simulation time.
    - つまりpredicted path上のcurrent simulation timeに一番近い直前のデータを始点として、remaining interval of timeでの動きを考えて、current simulation timeでの状態を予測する。



## Efficient Collision Detection Method

1. Represent each car as a set of circles
2. Check if a collision will occur between two circles

$$d_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \qquad r_i + r_j \geq d_{i,j}$$

3. Calculate collision point

$$C_X = \frac{(x_i * r_j) + (x_j * r_i)}{(r_i + r_j)}, \qquad C_Y = \frac{(y_i * r_j) + (y_j * r_i)}{(r_i + r_j)}$$

Efficient Collision Detection Method
- For the exact approach, a polygonal intersection is performed and any overlap implies a collision.
  - 福田さんのやり方exact方法の方だ。いいえ、そもそも四角形のpolygon intersection計算は簡単すぎるから。
- Circle approximation.
1. Represent each car as a set of circles.
   1. Which leads to 9 distance checks for 3 circle approximation, and 16 distance checks for 4 circle approximation.
   2. As the number of dynamic objects increases in the scene, the number of collision checks scales quadratically, if exhaustively evaluating all possible collisions.
      1. 対策：Only compare the ego vehicle to others, to keep computational complexity linear, or eliminate large portions of the pairwise comparisons based on a single point-to-point distance check between the centroids.
2. Check if a collision will occur between two circles: $d_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, $r_i + r_j \geq d_{i,j}$.
3. Calculate collision point: $C_x = \dfrac{(x_i * r_j) + (x_j * r_i)}{r_i + r_j}$, $C_y = \dfrac{(y_i * r_j) + (y_j * r_i)}{r_i + r_j}$.
   1. $r_i == r_j$の場合はすぐわかる。
- Tradeoff between accuracy and number of computations.
  - 単純にまるの数の話だ。

単語
- Evasive: If you take evasive action, you deliberately move away from someone or something in order to avoid meeting them or being hit by them.
- Geometry: The geometry of an object is its shape or the relationship of its parts to each other.
- Memory Footprint: Memory footprint refers to the amount of main memory that a program uses or references while running.
- Implication: The implications of something are the things that are likely to happen as a result.
- Bumper: Bumpers are bars at the front and back of a vehicle that protect it if it bumps into something.