Go, change the world®

ARTIFICIAL INTELLIGENCE IN AUTONOMOUS VEHICLES

VII semester
Department of AIML
RV College of Engineering

Course Incharge: Dr. Viswavardhan Reddy Karna

2024-2025

Course Contents

Go, change the world®

Semester: VII						
ARTIFICIAL INTELLIGENCE IN AUTONOMOUS VEHICLES **Category: Professional Core Elective** **(Theory)**						
Course Code	:	21AI73G1		CIE	:	100Marks
Credits: L: T: P	:	3:0:0		SEE	:	100 Marks
Total Hours	:	40T		SEE Duration	:	3.00 Hours

Unit-I	**8 Hrs.**
Introduction to Autonomous Driving: Autonomous Driving Technologies Overview, Autonomous Driving Algorithms, Autonomous Driving Client System, Autonomous Driving Cloud Platform Autonomous Vehicle Localization: Localization with GNSS, Localization with LiDAR and High-Definition Maps, Visual Odometry, Dead Reckoning and Wheel Odometry, Sensor Fusion	
Unit – II	8 Hrs.
Perception in Autonomous Driving: Introduction, Datasets, Detection, Segmentation, Stereo, Optical Flow, and Scene Flow, Tracking Deep Learning in Autonomous Driving Perception: Convolutional Neural Networks., Detection, Semantic Segmentation, Stereo and Optical Flow	
Unit –III	8 Hrs.
Prediction and Routing: Planning and Control Overview, Traffic Prediction, Lane Level Routing Decision, Planning, and Control: Behavioral Decisions, Motion Planning, Feedback Control	

Course Contents – Reference Books

Go, change the world®

Reference Books	
1.	Creating Autonomous Vehicle Systems, Second Edition Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot ,2nd Edition, September 2020, ISBN: ISBN: 9781681739366
2.	George Dimitrakopoulos, Aggelos Tsakanikas, Elias Panagiotopoulos, Autonomous Vehicles Technologies, Regulations, and Societal Impacts, 1st Edition,Elsevier Publications, 2021 , ISBN-10 1681730073
3.	Hanky Sjafrie, "Introduction to Self-Driving Vehicle Technology", 1st Edition, Published December 11, 2019 by Chapman and Hall/CRC, ISBN: 978-0-323-90137-6
4	Creating Autonomous Vehicle Systems Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot October 2017, ISBN-10 1681730073

Course Contents – Unit - II

Go, change the world®

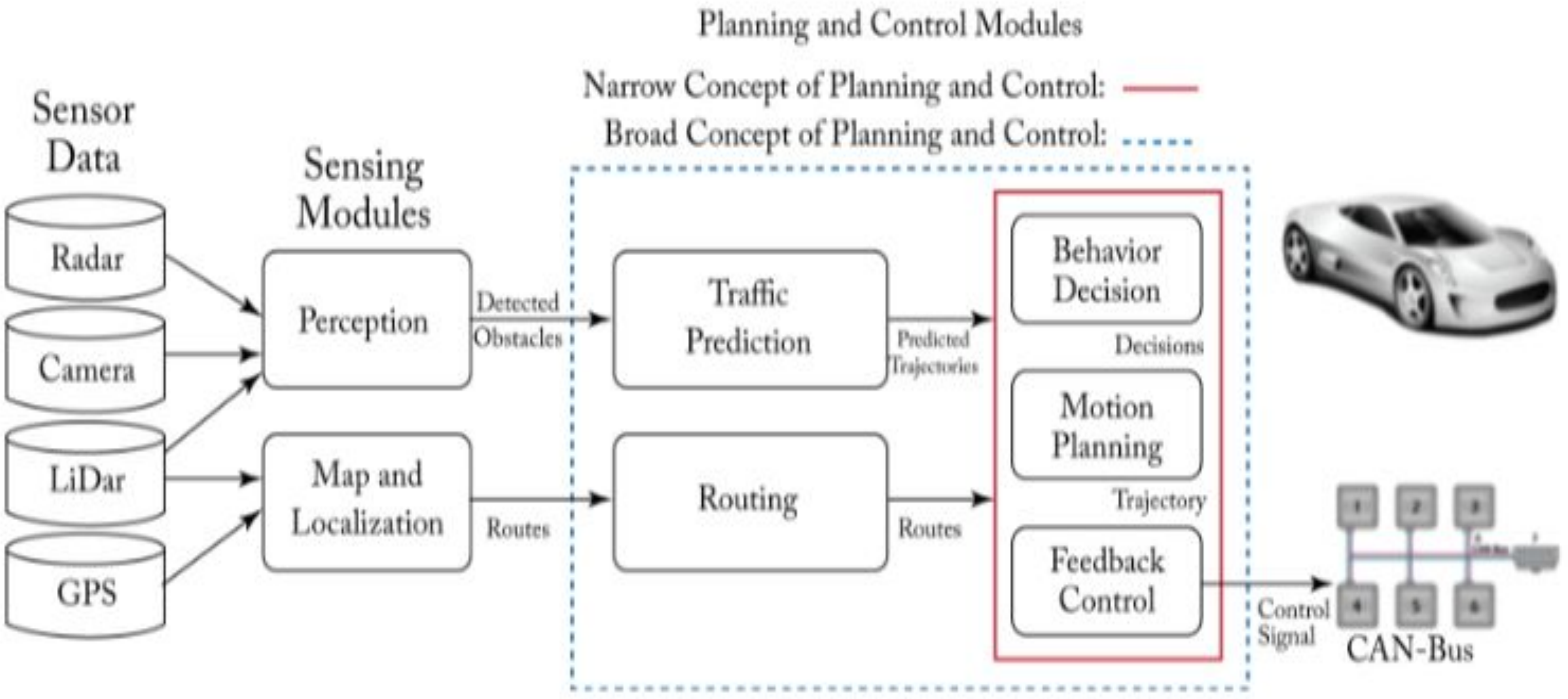
Prediction and Routing:

- Planning and Control Overview
- Traffic Prediction
- Lane Level Routing

Decision, Planning, and Control:

- Behavioral Decisions
- Motion Planning
- Feedback Control

ARCHITECTURE: PLANNING AND CONTROL IN A BROADER SENSE



Planning and control modules under narrow and broad concepts

ARCHITECTURE: Function of the Routing Module

Go, change the world®

- Issues with top-level navigation commands.
- Navigate from origin to destination by following a series of lanes.
- Key differences from traditional navigation services (e.g., Google Maps):
    - Relies on HD maps customized for autonomous driving.
    - Significantly more detailed and complex.

Overview of Traffic Prediction Module

- Inputs:
    - Perceived objects.
    - Attributes: position, velocity, and category (e.g., vehicle, cyclist, pedestrian).

- Outputs:
    - Predicted trajectories for perceived objects.
    - Trajectories include spatial and temporal information for downstream modules.

ARCHITECTURE: Function of the Routing Module

Implementation Approaches

- Traditional Approach:
  - Implemented as a peripheral software library.
  - Benefits: Simpler computation logic, stateless operation.

- Modern Approach:
  - Implemented as a standalone module.
  - Operates periodically, and maintains states for enhanced accuracy.
  - Importance has grown with industry evolution and real-world deployment.

ARCHITECTURE: Function of the Routing Module

Behavioral Decision Module

- Acts as the “co-pilot” of the autonomous system.

- Inputs:
  - Predicted trajectories from the traffic prediction module.
  - Commands from the routing module.

- Outputs:
  - Determines vehicle actions (e.g., follow, stop, yield).

  - Examples:
    - Follow the vehicle ahead in the current lane.
    - Stop at a traffic light and wait for pedestrians.
    - Yield to cross traffic at stop signs.

ARCHITECTURE: Function of the Routing Module

Go, change the world®

Introduction to Behavioral Decisions

- **Definition**:
  - Behavioral decisions encompass: Decisions for the autonomous vehicle itself.
    - Behavioral assessments of perceived or mapped objects.
- **Example**:
  - Detected vehicle in the same lane.
  - The routing module suggests remaining in the current lane.
  - Synthetic decision: Remain in the lane.
  - Individual decision: Follow the vehicle ahead.

- **Key Transition**:
  - Behavioral decisions convert into optimization constraints and costs in motion planning.

- **Synthetic Decision**:
  - Synthesized from individual decisions.
  - Crucial for determining end-state motion conditions.

- **Collaboration**:
  - Behavioral decision logic must align with motion planning.

ARCHITECTURE: Function of the Routing Module

Go, change the world®

Challenges and Scenarios

•**Command Set**:

- Explicit commands aid in debugging. Handles diverse traffic scenarios. Vague scenarios default to collision avoidance.

•**Worst-case Handling**:

- Implicit collision avoidance cost passed to motion planning.

•**Collaboration**: Behavioral decision logic must align with motion planning.

Motion Planning Overview

•**Definition**:

- Optimization problem for a local path from point A to point B.
- Attributes include location, heading, velocity, and curvature.

•**Consistency and Smoothness**:

- Trajectories should be: Consistent across planning cycles. Physically executable.

•**Inputs**:

- Mapping, localization, and perception feed directly into motion planning.

ARCHITECTURE: Function of the Routing Module

Collaboration of Modules

- **Redundant Inputs**:
    - Mapping, localization, and perception provide backups for prediction.

- **New Obstacles**: Redundancy ensures collision avoidance.

- **Feedback Control**: Converts trajectory points into drive-by-wire signals.

Feedback Control Module

- **Core Task**:
    - Drive-by-wire signals for brakes, steering, and gas.

- **Execution**:
    - Ensures vehicle path conforms to planned trajectory.
    - Considers the physical model of the vehicle and road.

Traffic Prediction in Autonomous Vehicles

- **Definition**:
  - Traffic prediction forecasts the behavior of detected objects in the near future.
  - Outputs spatial-temporal trajectory points for downstream modules.

- **Attributes of Detected Obstacles**:
  - Position, velocity, heading, and acceleration.
  - Immediate predictions consider simple physical rules.

Behavioral-Level Prediction

- **Objective**:
  - Predict behavior over a few seconds, considering:
    - Historical behavior.
    - Surrounding scenarios.
    - Map features.

- **Example**:
  - At a traffic junction:
    - Will a vehicle go straight or turn?
    - Will a pedestrian cross or remain still?

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Machine Learning in Traffic Prediction

- **Behavioral Prediction**:
  - Classification problems solved using machine learning.
  - Examples:
    - Lane changes. Intersection crossing decisions.

- **Outputs**: Predicted trajectories with timing, speed, and headings.

Sub-Problems in Traffic Prediction

- **Classification Problem**:
  - Predict categorical behaviors:
    - Will a vehicle change lanes or remain in its lane?
    - Will a pedestrian cross an intersection?

- **Regression Problem**:
  - Generate precise paths with speed and timing.
  - Examples:
    - Pedestrian maintaining steady speed while crossing.
    - Vehicle decelerating during a turn before accelerating.

Traffic Prediction in Autonomous Vehicles

Practical Challenges and Considerations

•**Multiple Factors**:

- Predictions must integrate: Physical attributes. Behavioral trends. Environmental context.

•**Dynamic Scenarios**: Need to adapt to new obstacles and prediction failures.

•**Collaboration**: Seamless interaction with downstream modules for motion planning and control.

Applications in Motion Planning

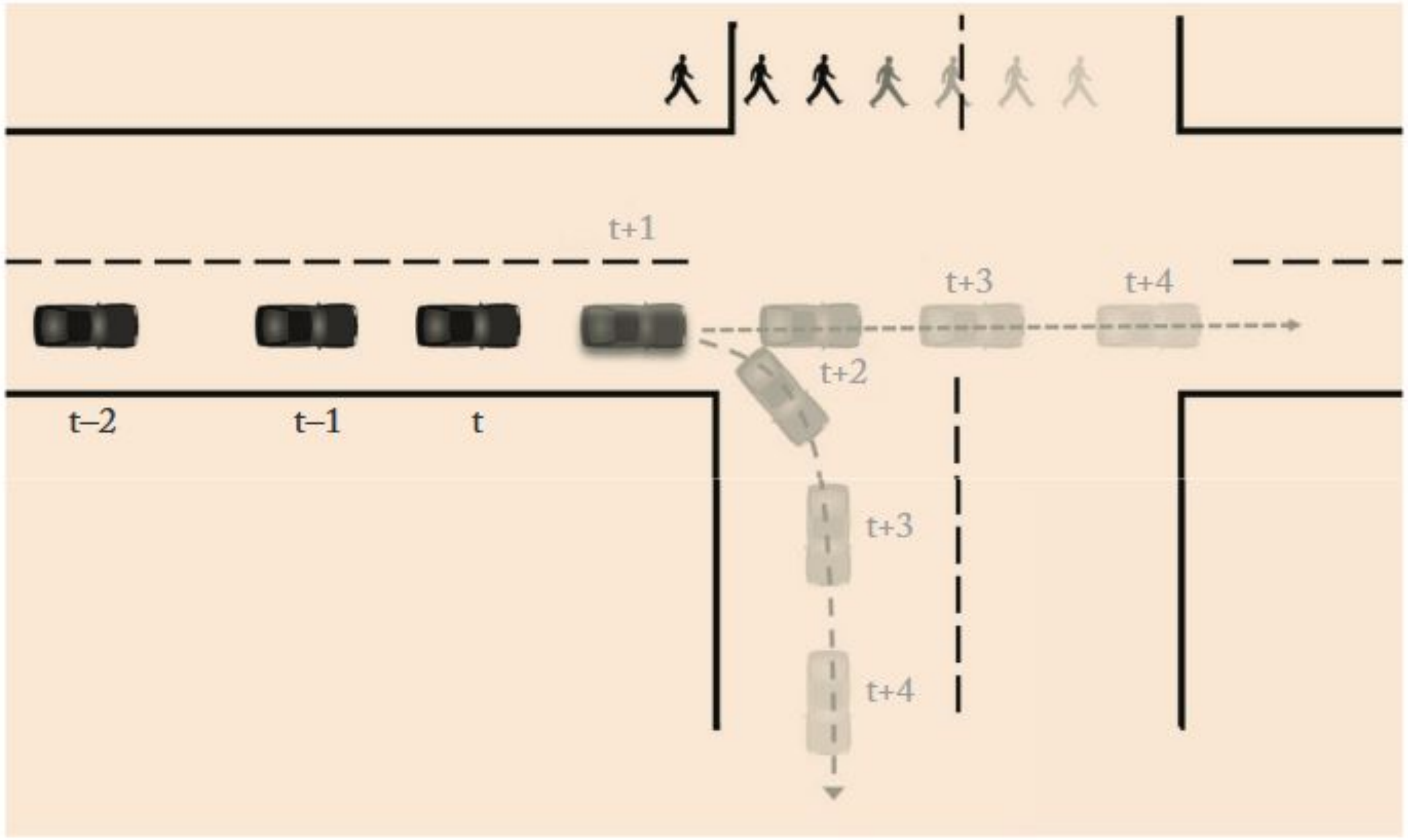
•**Role**: Provides constraints and goals for motion planning.

•**Integration**:

- Predictions ensure safe and smooth trajectories.
- Helps anticipate changes in the environment.

Traffic Prediction in Autonomous Vehicles

Go, change the world®



Traffic prediction for objects on the road

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Behavioral Prediction for Road Objects

•**Definition**:

- Behavioral prediction depends on the type of road object.
- Focus: Vehicle behavior (excludes pedestrians and cyclists for simplicity).

•**Objective**:

- Predict vehicle behaviors such as:Remain in current lane. Make a turn. Change lanes.

Challenges in Vehicle Behavioral Prediction

•**Complexity**:

- Real-world scenarios add layers of complexity:
  - Multiple turn lanes.
  - Non-standard intersections (not always four-way).
  - Lanes naturally yielding into turns.

•**Scalability Issues**:

- Simple action-based models (e.g., remain, switch, turn) are not sufficient.
- Map-based categories result in an overwhelming number of labels.

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Proposed Solution

•**Decoupling Behavior from Maps**:
- Define behavior as whether a vehicle will follow a finite set of lane sequences.
- Avoids dependence on specific map scenarios.

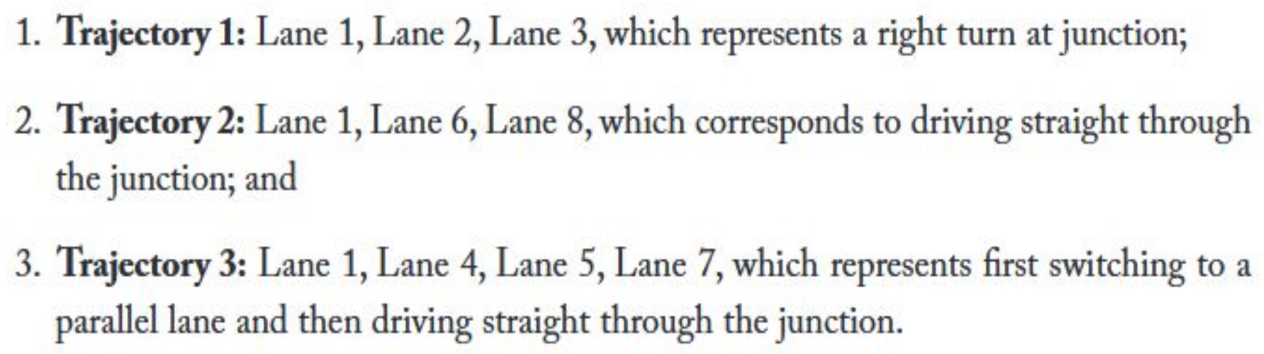
•**Assumptions**: Vehicles generally adhere to lanes on a map.

•**Lane Sequences**: A vehicle's trajectory can be described as a series of possible lane sequences.

Lane Sequence Prediction

•**Key Approach**: Use current and historical data to predict lane sequences.

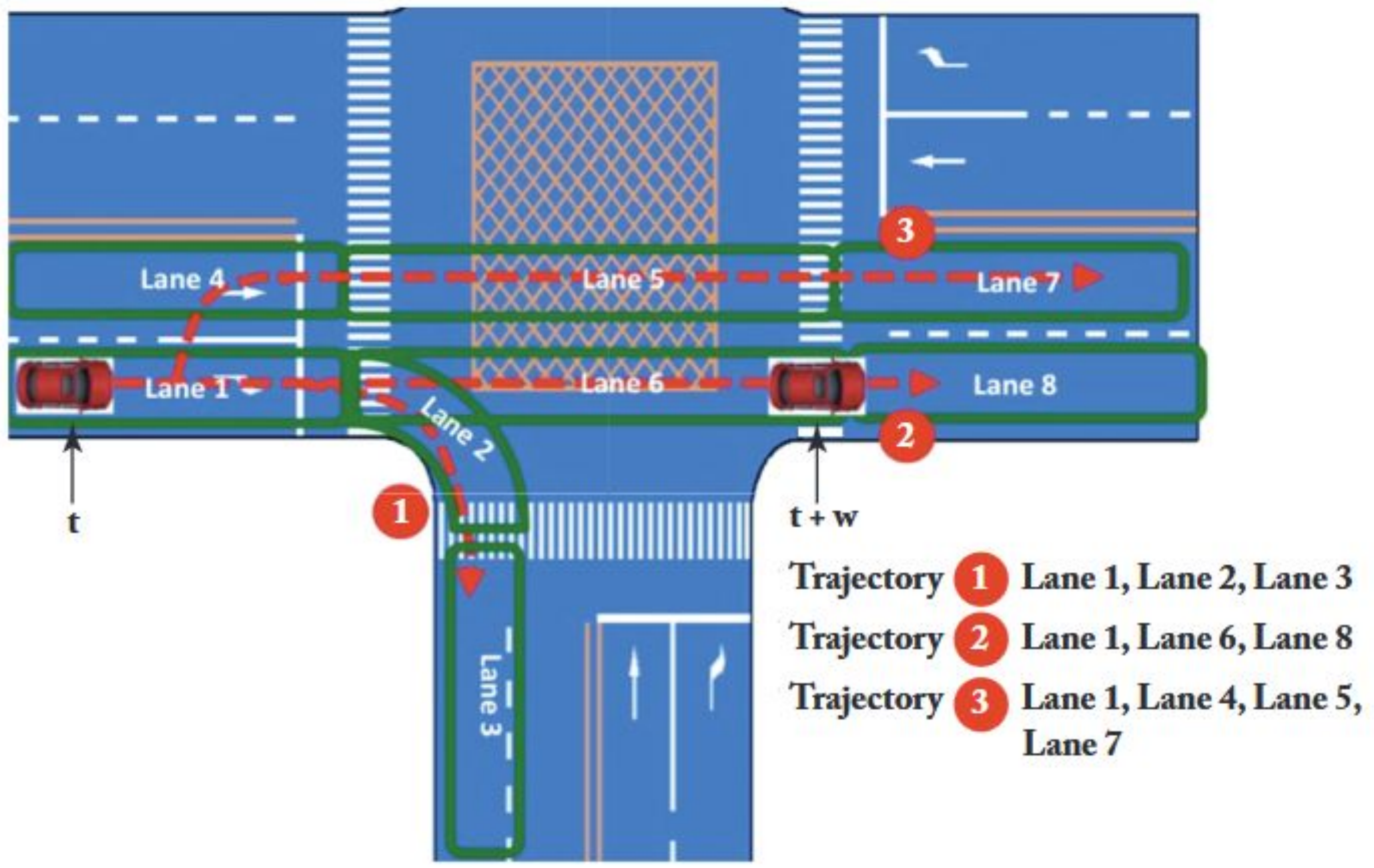
•**Example:**
- Vehicle in Lane 1 at time t.
- Three possible legal lane sequences:
  - Sequence 1: Lane 1 → Lane 2 → Lane 3.
  - Sequence 2: Lane 1 → Lane 4.
  - Sequence 3: Lane 1 → Lane 5 → Lane 6.



1. **Trajectory 1:** Lane 1, Lane 2, Lane 3, which represents a right turn at junction;
2. **Trajectory 2:** Lane 1, Lane 6, Lane 8, which corresponds to driving straight through the junction; and
3. **Trajectory 3:** Lane 1, Lane 4, Lane 5, Lane 7, which represents first switching to a parallel lane and then driving straight through the junction.

Traffic Prediction in Autonomous Vehicles

Go, change the world®


Lane 4
Lane 5
Lane 7
Lane 1
Lane 6
Lane 8
Lane 2
Lane 3
1
2
3
t
t + w
Trajectory 1 Lane 1, Lane 2, Lane 3
Trajectory 2 Lane 1, Lane 6, Lane 8
Trajectory 3 Lane 1, Lane 4, Lane 5, Lane 7

Formulating a behavioral traffic prediction problem as a binary classification problem with regard to lane sequences.

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Machine Learning Model Customization

- **Customization by Object Type**:
    - Separate models for vehicles, pedestrians, and cyclists.
    - Focus on vehicles for this presentation.

- **Behavior Prediction Model**:
    - Classification problem: Predict the most likely lane sequence.
    - Regression problem: Estimate timing and speed along the sequence.

Applications and Advantages

- **Applications**:
    - Improved motion planning and collision avoidance.
    - Enhanced predictions for complex traffic scenarios.

- **Advantages**:
    - Scalable and map-agnostic behavior classification.
    - Works across diverse intersections and road configurations.

Traffic Prediction in Autonomous Vehicles

Feature Design for Vehicle Behavior Prediction

- **Definition**:
  - Feature engineering involves using domain knowledge to extract features from raw data for machine learning classifiers.

- **Focus**:
  - Applying feature engineering to the vehicle behavioral prediction problem.

- **Categories of Features**:
  - Vehicle history features.
  - Lane sequence features.
  - Surrounding object features.

Traffic Prediction in Autonomous Vehicles

Vehicle History Features

•**Definition**: Represents historical movement of the vehicle over a window of *w* frames.

•**Key Attributes**:

- Absolute position of the vehicle. Relative position to the lane.

•**Purpose**:

- Reflects how the vehicle has moved along current or previous lanes.

Lane Sequence Features

•**Definition**: Attributes of the designated lane sequence to be classified as "will take" or "will not take.“

•**Key Attributes**:

- Sampled lane points with features such as:
  - Heading.
  - Curvature.
  - Distance to lane boundary.

•**Purpose**:

- Represents the shape of the lane sequence to be taken.

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Surrounding Object Features

- **Definition**: Captures objects surrounding the vehicle whose behavior needs prediction.

- **Challenges**:
  - Computational complexity.
  - Integration of object dynamics.

- **Example**:
  - For adjacent lanes, project the target vehicle and compute forward distances to adjacent objects.

- **Purpose**:
  - Considers interactions with surrounding objects in behavior prediction.

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Models for Vehicle Behavioral Prediction

- The feature set is not exhaustive but provides feasible suggestions.
- Features can be adapted to specific machine learning models.

Memory-less Models

- **Definition**: Examples: SVM, DNN. The model remains unchanged after training. Output does not depend on previous inputs.
- **Historical Information**:
  - Must be explicitly encoded into features.
  - Example: Use multiple historical frames for training and prediction.

- **Feature Set:** Considers vehicle historical information.

Memory Models

- **Definition**: Examples: LSTM, RNN. Output depends on previous inputs (has memory).

- **Training**: More difficult to train compared to memory-less models.

- **Input Requirements**:
  - Current frame information is sufficient. Historical inputs are memorized by model parameters.

Traffic Prediction in Autonomous Vehicles

Go, change the world®

Model Selection Criteria

•**Scenarios for Memory-less Models**:
- Suitable for simple mapping and environments.

•**Scenarios for Memory Models**: Recommended for complex traffic conditions.

•**Implementation**:
- Memory Models: These are easier to implement online, as only current information is required, and the model itself memorizes the historical information.
- Memory-less Models: Require online systems to store and process historical data. (features)

Time Window Considerations

•**Historical Data**:
- "w" represents the time window for tracking objects.
- Typical value: 5 seconds, though shorter windows like 3 seconds are more reliable.

•**Predicted Trajectory**:
- Should cover minimum distance or time.
- Example: 3 to 5 seconds prediction.

Traffic Prediction in Autonomous Vehicles

Go, change the world®

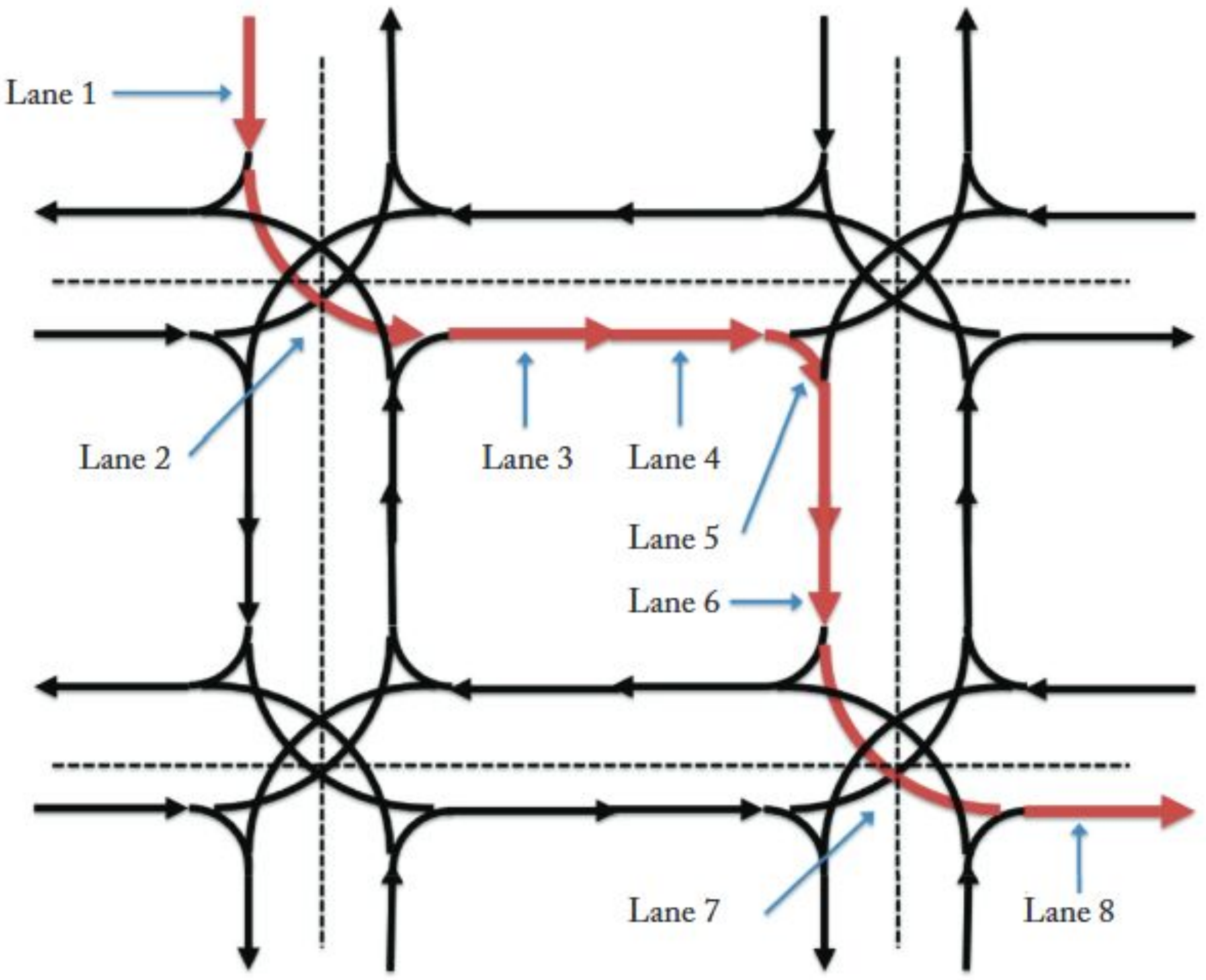
Metrics in Behavioral Prediction

- **Key Metrics**:
  - Precision: Proportion of predicted trajectories actually taken by the object.
  - Recall: Proportion of actual behavioral trajectories that were predicted.

- **Evaluation**:
  - Metrics are computed by aggregating predictions across all frames.

LANE LEVEL ROUTING

Go, change the world®



igure 5.5: Routing output on lane levels defined by the HD map.

- The diagram leverages HD maps, which are highly detailed and precise maps used in autonomous navigation.

- These maps capture fine-grained details like lane divisions, curvature, and traffic flow constraints.

Turn Management and Lane Change:

- At intersections, the routing path transitions between lanes. For example, the path makes a turn at the intersection between Lane 1 and Lane 2.

- Lane changes are also evident, which are critical for maintaining correct routing while adhering to traffic rules.

LANE LEVEL ROUTING

Go, change the world®

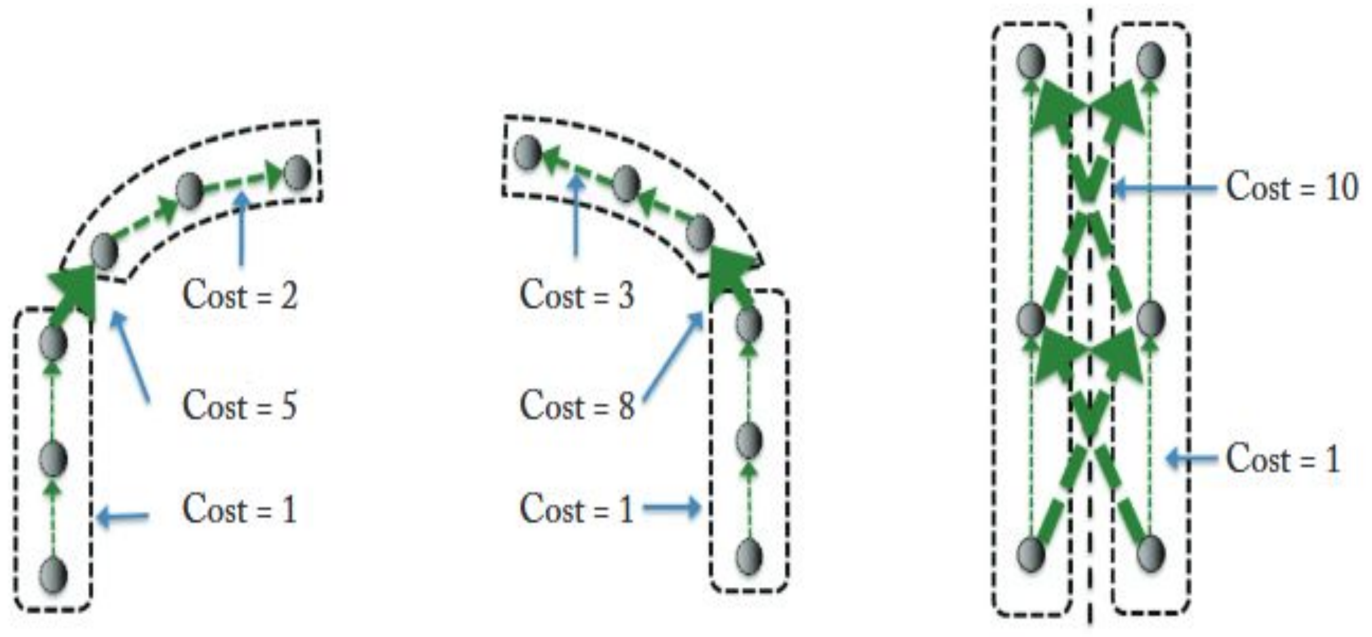


Figure 5.6: Costs of edges connecting lane points under scenarios of Right Turn, Left Turn, and Switch Lanes.

Lane Points:

- The gray circles represent lane points (nodes) in the road network.
- These are connected by green arrows (edges), representing possible transitions between nodes.

Costs on Edges:

- Each edge (green arrow) has a labeled cost, indicating the difficulty or effort required to transition between two nodes.
- Lower costs represent easier transitions, while higher costs indicate transitions that are less desirable or more complex.

LANE LEVEL ROUTING

Go, change the world®

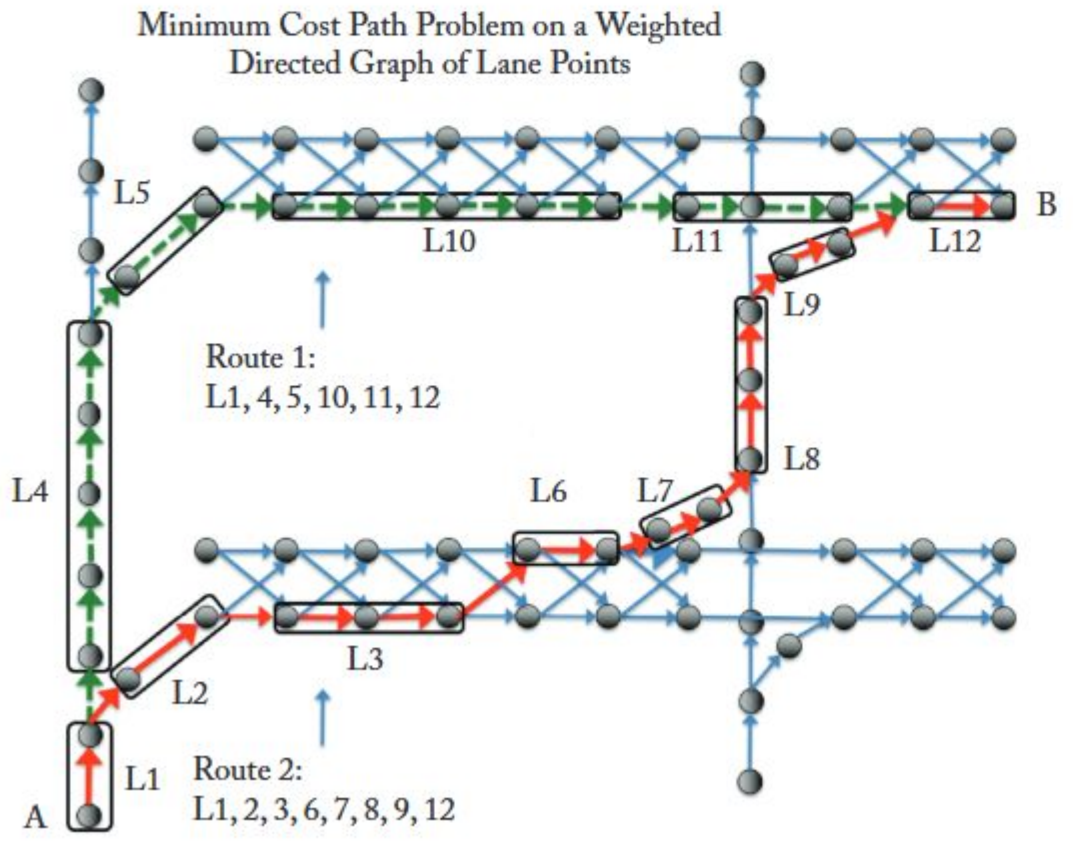


Figure 5.7: Routing as a minimum cost path problem on lane points connected weighted graph.

Lane Points (Nodes):

- The gray circles represent lane points, forming a graph of interconnected nodes. These correspond to specific locations on the road.

Edges and Weights:

- The connections (arrows) between nodes represent possible paths or transitions between lane points.

- Each edge is part of a **weighted directed graph**, meaning transitions have a cost (weight), which influences the path selection.

Routes:

•Two distinct routes, **Route 1** and **Route 2**, are highlighted. Each route connects point **A** (start) to **B** (destination) while minimizing the cumulative cost.

•Route 1 (green path): Nodes L1, 4, 5, 10, 11, 12.
•Route 2 (red path): Nodes L1, 2, 3, 6, 7, 8, 9, 12.

TYPICAL ROUTING ALGORITHMS

Go, change the world®

```
**function** Dijkstra_Routing(*LanePointGraph(V,E)*, *src*, *dst*)
    create vertex set $Q$
    create map *dist*, *prev*
    **for each** lane point $v$ in $V$:
        $dist[v]$ = **inf**
        $prev[v]$ = **nullptr**
        add $v$ to $Q$
    $dist[src]$ = **0**
    **while** $Q$ is **not empty**:
        $u$ = vertex in $Q$ s.t. $dist[u]$ is the **minimum**
        remove $u$ from $Q$
        **for each** connected lane point $v$ of $u$:
            $candidate = dist[u] + cost(u, v)$
            **if** $candidate < dist[v]$:
                $dist[v] = candidate$
                $prev[v] = u$;
    $ret$ = **empty sequence**
    $u = dst$
    **while** $prev[u]$ != **nullptr**:
        insert $u$ at the beginning of $ret$
        $u = prev[u]$
    insert $u$ at the beginning of $ret$
    merge lane point in $ret$ with same lane id and return the merged sequence
```

: Dijkstra implementation of routing based on weighted directed graph of lane points.

TYPICAL ROUTING ALGORITHMS

Go, change the world®

```
function AStar_Routing(LanePointGraph(V,E), src, dst)
    create vertex set closedSet          // set of already visited nodes
    create vertex set openSet            // set of nodes to be expanded
    insert src into openSet
    create map gScore, fScore with default value inf
    create prev_map with default value nullptr
    fScore[src] = h(src, dst)
    while openSet is not empty:
        current = the node v in openSet s.t. fScore[v] is minimum in openSet
        if current = dst
            return reconstruction_route(prev_map, current)
        remove current from openSet
        insert current into closedSet
        for each neighbor u of current:
            if u is in closedSet:
                continue;  // ignore the neighbor who has already been evaluated
            candidate_score = gScore[current] + h(current, u)
            if u not in openSet:          // discovered a new node
                insert u into openSet
            else if candidate_score >= gScore[u]:   // this is not a better path
                continue;
            prev[u] = current
            gScore[u] = candidate_score
            fScore[u] = gScore[u] + h(u, dst)
```

: A* Algorithm on autonomous vehicle routing.

ROUTING GRAPH COST: WEAK OR STRONG ROUTING

Go, change the world®

Configuration of Costs > Algorithm Choice

- Adjusting costs between lane points is critical for a functioning routing module.
- Example:
    - High costs for congested roads to avoid them.
    - Infinite costs for traffic-controlled roads to make them less preferred.

Dynamic Cost Adjustments

- **Dynamic Updates:**
    - Costs are adjusted based on:
        - Real-time traffic data.
        - Lane preferences.
    - Reflects evolving road conditions.
- **Goal:**
    - Optimize lane selection and improve routing efficiency.

ROUTING GRAPH COST: WEAK OR STRONG ROUTING

Go, change the world®

Handling Large Road Graphs

Process:

1. Pre-load road graph data.
2. Construct lane point graph ad-hoc.
3. Steps:
    1. If the route isn’t found within a small radius, load a larger radius of road graph.
    2. Reconstruct the lane point graph.
    3. Re-compute the route.

Types of Routing Requests

Key Types:

1. **Passenger-Initiated Routing:**
    2. User sets source and destination.
2. **Downstream Module Routing:**
    2. Triggered by Behavior Decision or Motion Planning modules.

ROUTING GRAPH COST: WEAK OR STRONG ROUTING

Go, change the world®

Strong vs. Weak Routing

- **Strong Routing:**
  - Downstream modules strictly follow routing outputs.
  - Example: Vehicle follows designated lanes strictly. Sends re-routing requests if deviations are needed.

- **Weak Routing:** Allows flexibility to adapt to scenarios.
  - Example: Vehicle can switch lanes to overtake slow-moving vehicles.

Example Scenario

Strong vs. Weak Routing in Action:

- **Scenario:**
  - Routing suggests staying in the current lane.
  - A slow vehicle is ahead.
- **Strong Routing:**
  - Reduce speed and follow the slow vehicle.
- **Weak Routing:**
  - Switch to the adjacent lane, overtake, and merge back.

Behavioral Decisions

Go, change the world®

- The behavior decision module is the “co-driver” in the general autonomous vehicle planning and control modules.

- It is the module where most data sources are consumed and processed.

- Data sources fed to the decision module include but are not limited to, information about the autonomous vehicle, including location, speed, velocity, acceleration, heading, current lane info, and any surrounding objects information within a certain radius.

- The mission of the behavioral decision module is to compute the Behavioral level decision given all these various input data sources.

Behavioral Decisions - MARKOV DECISION PROCESS APPROACH

- A Markov Decision Process (MDP) is defined by the following five-element tuple (S, A, Pa, Ra, γ),

- where: S represents the state space of the autonomous vehicle.

- A represents the behavioral decision output space, for instance, a decision state could be to Follow a vehicle on the current lane, Switch Lanes to an adjacent parallel lane, Turn Left/
- Right, Yield, or Over-take a crossing vehicle at a junction, Stop for traffic lights and pedestrians, etc

- P_a (s, s') = P (s' |s, a) is the conditional probability to reach state s', given that the vehicle is currently at state s and takes action a;

- R_a (s, s') is the reward function for transforming from state s to state s' by taking action
- a.

Behavioral Decisions - MARKOV DECISION PROCESS APPROACH

- γ is the decay factor for the reward. The reward at the current time has a factor of 1 while the reward for the next time frame will be discounted by a factor of γ.

- Accordingly, the reward for t time frames in the future will be deemed as γt for the time being.

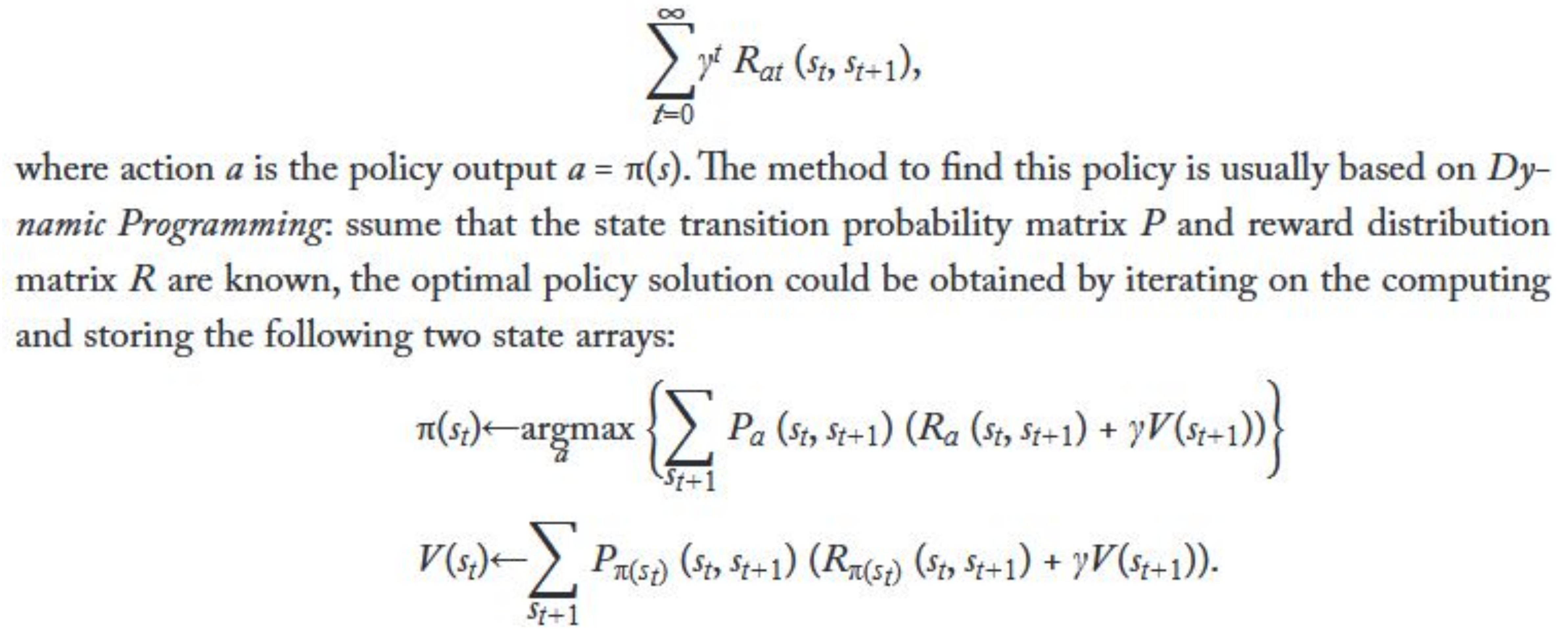
- With the formal MDP setting, the problem that behavioral decision needs to solve, is to find an optimal policy π to perform the mapping S→A

- The behavioral decision policy π is to optimize the accumulated rewards from the present time to the future.

- Mathematically, the accumulated reward to maximize is written as

Behavioral Decisions - MARKOV DECISION PROCESS APPROACH

Go, change the world®



$$\sum_{t=0}^{\infty} \gamma^t R_{at}(s_t, s_{t+1}),$$

where action a is the policy output $a = \pi(s)$. The method to find this policy is usually based on *Dynamic Programming*: ssume that the state transition probability matrix P and reward distribution matrix R are known, the optimal policy solution could be obtained by iterating on the computing and storing the following two state arrays:

$$\pi(s_t) \leftarrow \underset{a}{\operatorname{argmax}} \left\{ \sum_{s_{t+1}} P_a(s_t, s_{t+1}) \left(R_a(s_t, s_{t+1}) + \gamma V(s_{t+1}) \right) \right\}$$

$$V(s_t) \leftarrow \sum_{s_{t+1}} P_{\pi(s_t)}(s_t, s_{t+1}) \left(R_{\pi(s_t)}(s_t, s_{t+1}) + \gamma V(s_{t+1}) \right).$$

V (st) represents the accumulated future rewards discounted at the current time, and π(st) represents the policy that we want to search for. The solution is based on repeatedly iterating between possible state pairs (s, s'), until the above two state arrays converge

Behavioral Decisions - MARKOV DECISION PROCESS APPROACH

Go, change the world®

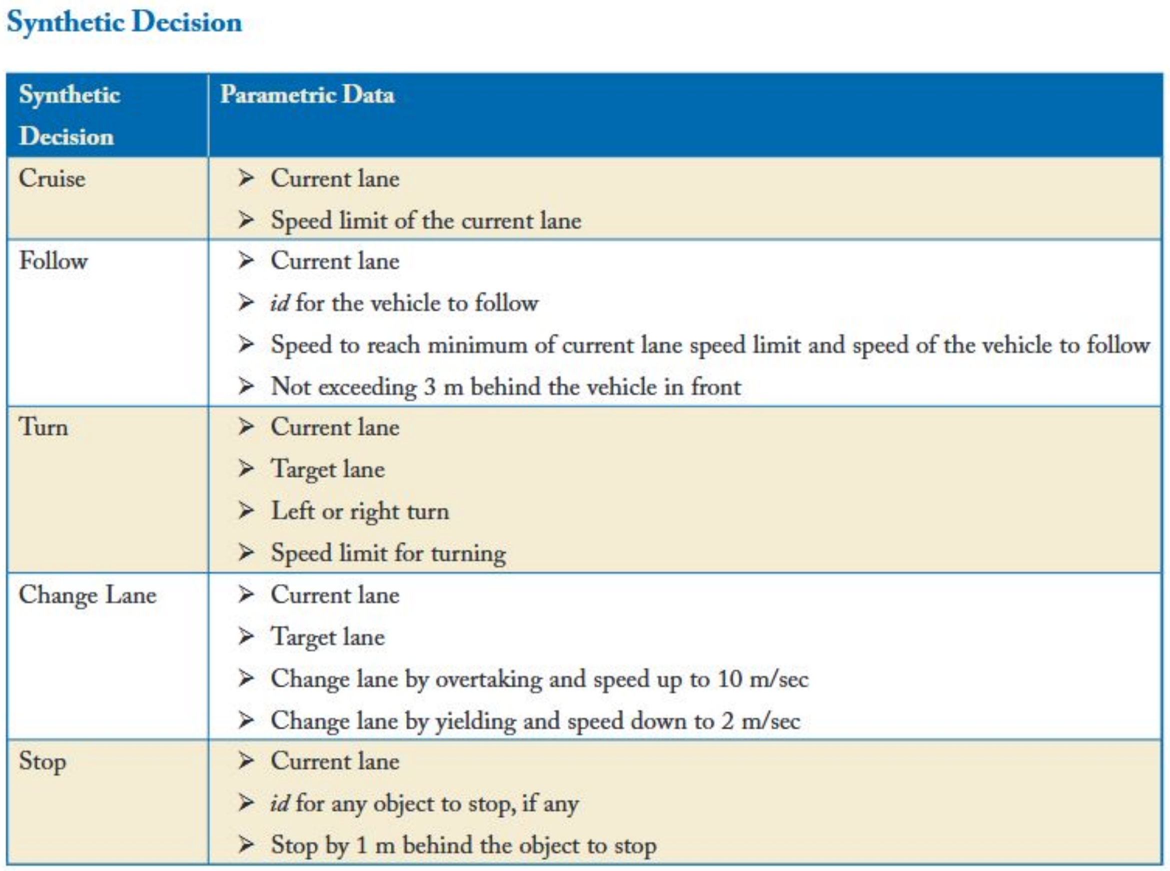
A good reward function in an MDP-based decision module should exhibit the following characteristics.

- It should be able to reach each destination: incentivize the autonomous vehicle to follow the routing module output route to reach the destination
- It should be safe and collision-free: If the state is based on N × N equal square grids centered on the vehicle, then any decision to move to another grid where collision might occur should be negatively rewarded. Movements to grids with a lower collision possibility or larger distance to collision likely grids should be rewarded.
- The ride should be comfortable and smooth

Behavioral Decisions - SCENARIO-BASED DIVIDE AND CONQUER APPROACH

Go, change the world®

- Divide and Conquer to decompose the vehicle's surroundings into scenarios.
- compute an Individual Decision for each object
- Synthetic Decision for the vehicle itself is computed by consolidating all the individual decisions

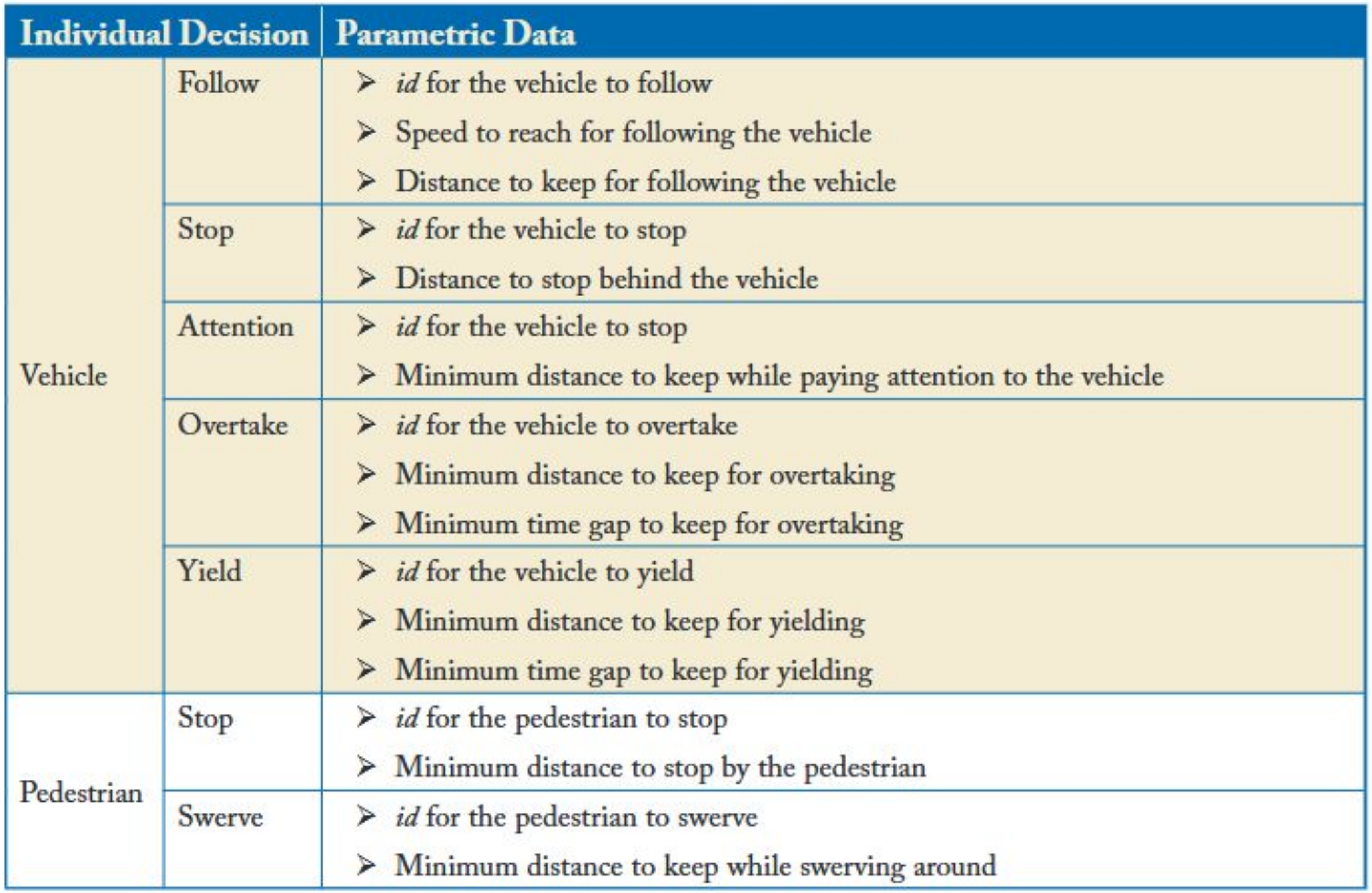


Synthetic Decision

Synthetic Decision	Parametric Data
Cruise	➢ Current lane ➢ Speed limit of the current lane
Follow	➢ Current lane ➢ *id* for the vehicle to follow ➢ Speed to reach minimum of current lane speed limit and speed of the vehicle to follow ➢ Not exceeding 3 m behind the vehicle in front
Turn	➢ Current lane ➢ Target lane ➢ Left or right turn ➢ Speed limit for turning
Change Lane	➢ Current lane ➢ Target lane ➢ Change lane by overtaking and speed up to 10 m/sec ➢ Change lane by yielding and speed down to 2 m/sec
Stop	➢ Current lane ➢ *id* for any object to stop, if any ➢ Stop by 1 m behind the object to stop

Figure 6.1: Synthetic decision with its parameters in behavioral decision.

Behavioral Decisions - SCENARIO-BASED DIVIDE AND CONQUER APPROACH

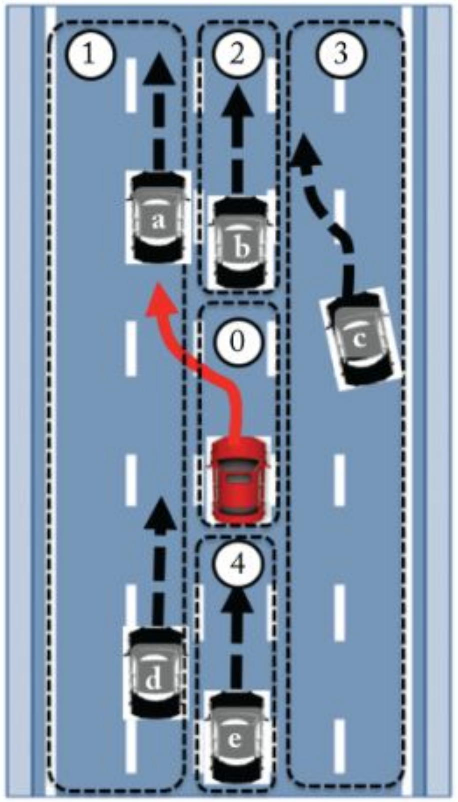


Individual Decision		Parametric Data
Vehicle	Follow	➢ *id* for the vehicle to follow ➢ Speed to reach for following the vehicle ➢ Distance to keep for following the vehicle
	Stop	➢ *id* for the vehicle to stop ➢ Distance to stop behind the vehicle
	Attention	➢ *id* for the vehicle to stop ➢ Minimum distance to keep while paying attention to the vehicle
	Overtake	➢ *id* for the vehicle to overtake ➢ Minimum distance to keep for overtaking ➢ Minimum time gap to keep for overtaking
	Yield	➢ *id* for the vehicle to yield ➢ Minimum distance to keep for yielding ➢ Minimum time gap to keep for yielding
Pedestrian	Stop	➢ *id* for the pedestrian to stop ➢ Minimum distance to stop by the pedestrian
	Swerve	➢ *id* for the pedestrian to swerve ➢ Minimum distance to keep while swerving around

Figure 6.2: Individual decisions with parameters in behavior decision module.

Behavioral Decisions - SCENARIO-BASED DIVIDE AND CONQUER APPROACH

Go, change the world®



Synthetic Decision:
Switch lane from the current lane to theleft lane: yield to vehicle **a**, overtake vehicle **d**, and pay attention to vehicle **b** in the current lane.

Scenarios and Individual Decisions:
1. Master Vehicle
2. Left Lane(s); Overtake **d** and yield to **a**
3. Front Vehicle(s): Pay attention to **b**
4. Right Lane(s): Ignore **c**
5. Rear Vehicle(s): Ignore **e**

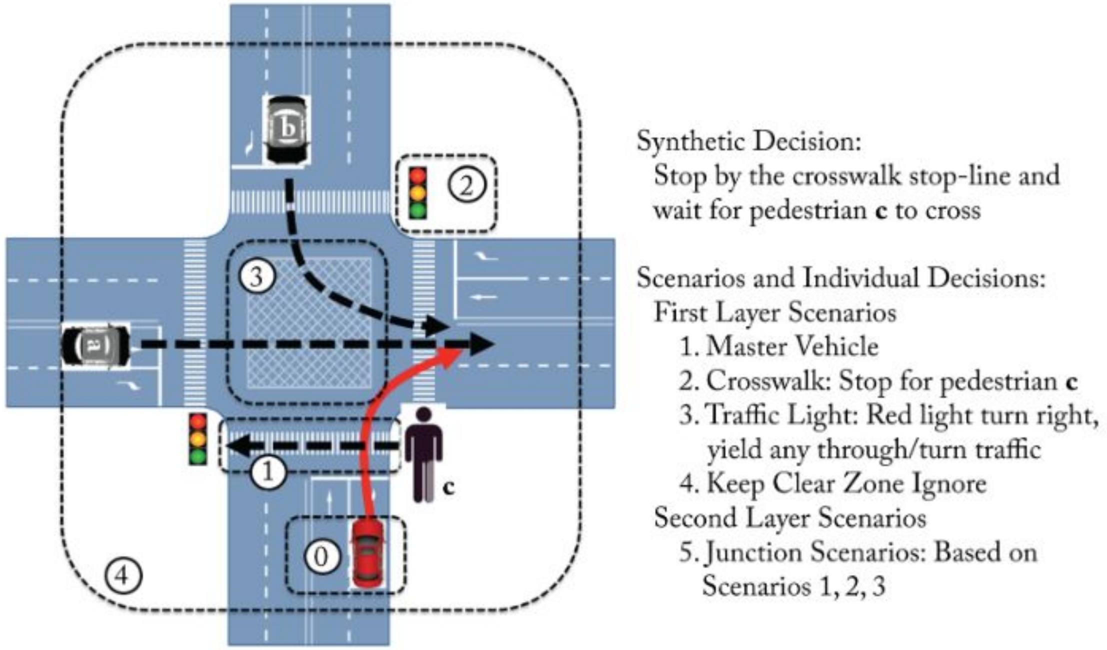
Figure 6.3: (a) Layered scenarios while performing a lane switch.

Key Elements:

1. **Master Vehicle (Red Car)**: The central vehicle executing the lane switch to the left lane.
2. **Surrounding Vehicles**:
   - **Vehicle a**: Located in the left lane ahead of the Master Vehicle.
   - **Vehicle b**: In the current lane, directly ahead of the Master Vehicle.
   - **Vehicle c**: In the right lane, not directly relevant to the lane-switching maneuver.
   - **Vehicle d**: In the left lane, behind the Master Vehicle.
   - **Vehicle e**: In the current lane, behind the Master Vehicle.

Behavioral Decisions - SCENARIO-BASED DIVIDE AND CONQUER APPROACH

Go, change the world®



Synthetic Decision:
Stop by the crosswalk stop-line and wait for pedestrian **c** to cross

Scenarios and Individual Decisions:
First Layer Scenarios
1. Master Vehicle
2. Crosswalk: Stop for pedestrian **c**
3. Traffic Light: Red light turn right, yield any through/turn traffic
4. Keep Clear Zone Ignore

Second Layer Scenarios
5. Junction Scenarios: Based on Scenarios 1, 2, 3

Figure 6.3: (b) Layered scenarios while at junction.

Key Elements:

1. **Master Vehicle (Red Car)**: The central vehicle executing the lane switch to the left lane.
2. **Surrounding Vehicles**:
   - **Vehicle a**: Located in the left lane ahead of the Master Vehicle.
   - **Vehicle b**: In the current lane, directly ahead of the Master Vehicle.
   - **Vehicle c**: In the right lane, not directly relevant to the lane-switching maneuver.
   - **Vehicle d**: In the left lane, behind the Master Vehicle.
   - **Vehicle e**: In the current lane, behind the Master Vehicle.

Behavioral Decisions - SCENARIO-BASED DIVIDE AND CONQUER APPROACH

Go, change the world®

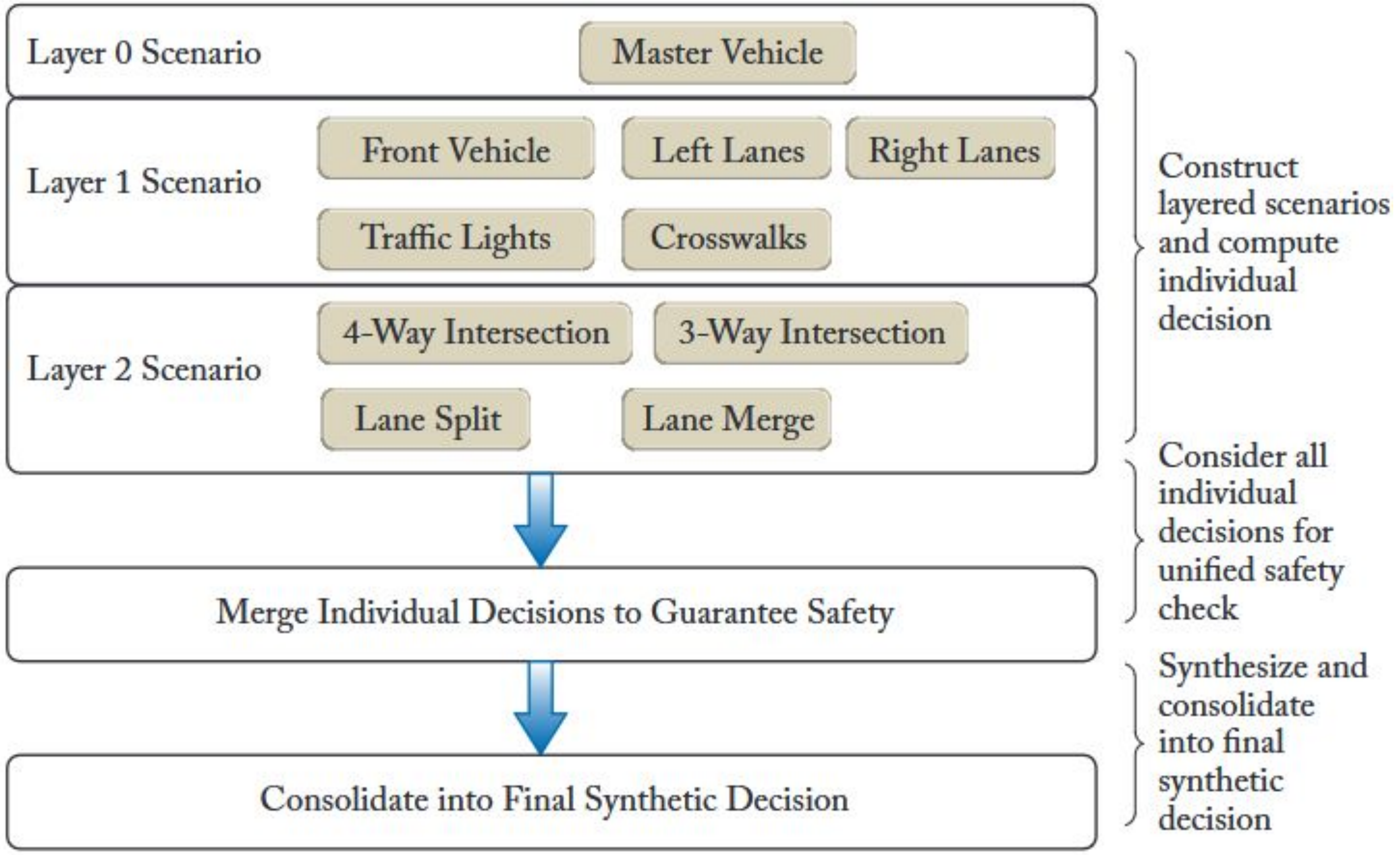

Layer 0 Scenario
Master Vehicle
Layer 1 Scenario
Front Vehicle
Left Lanes
Right Lanes
Traffic Lights
Crosswalks
Layer 2 Scenario
4-Way Intersection
3-Way Intersection
Lane Split
Lane Merge
Merge Individual Decisions to Guarantee Safety
Consolidate into Final Synthetic Decision
Construct layered scenarios and compute individual decision
Consider all individual decisions for unified safety check
Synthesize and consolidate into final synthetic decision

Figure 6.4: Architecture of a rule-based behavior decision system containing layered scenarios.

Components and Process:
Layer 0 Scenario (Master Vehicle):

- Represents the autonomous vehicle itself, which is at the center of the decision-making process.
- The system begins by assessing the Master Vehicle's current state and position in the driving environment.