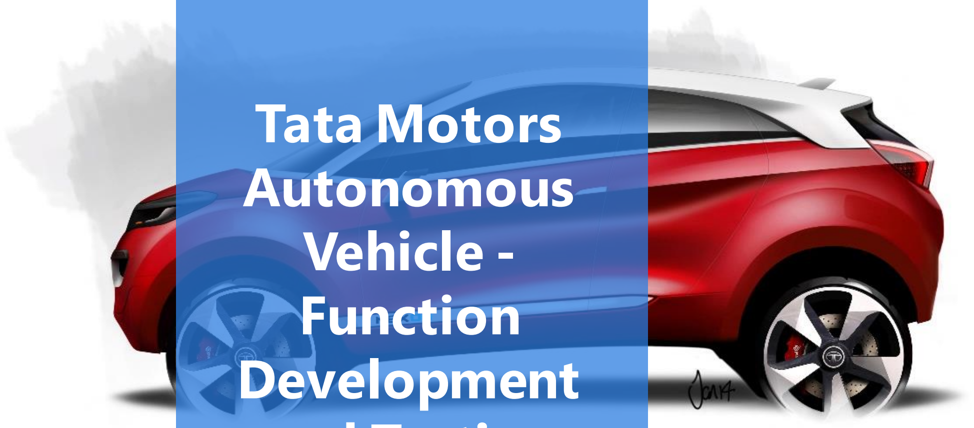
TATA MOTORS
EUROPEAN TECHNICAL CENTRE

Tata Motors Autonomous Vehicle - Function Development and Testing



Dr. Mark Tucker

TATA MOTORS EUROPEAN TECHNICAL CENTRE

Overview

- Introduction
  - Tata/TMETC
  - Autonomy
  - UK Autodrive
- TMETC Autonomous Hexa Architecture
  - Functional/Hardware/Software
- Autonomous Functions (Sensing/Perception/Planning/Control)
  - Global/Behaviour/Trajectory Planning
  - Control Options
  - Model Predictive Control
- Experiences
  - Time Issues
  - Data Logging and Visualisation
- Conclusion

TATA MOTORS
Connecting Aspirations

TATA

The Tata Group

Operations in more than 100 countries

660,000 employees

Tata Group $100bn turnover

Tata Motors $42bn turnover

Tata Motors European Technical Centre

- Created 2005
- Based in Coventry
- Wholly-owned subsidiary of Tata Motors
- Research & development principally for Tata Motors
- Engineering Centre, Design Studio, Workshops
- 180-strong workforce

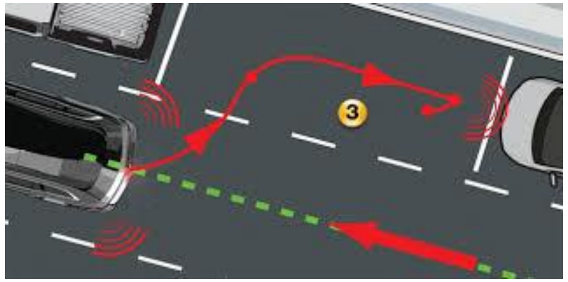
Autonomy and ADAS Terminology

Advanced Driver Assistance Systems (ADAS)

AEB
(Automatic Emergency Braking)

LDW
(Lane Departure Warning)

ACC
(Automatic Cruise Control)

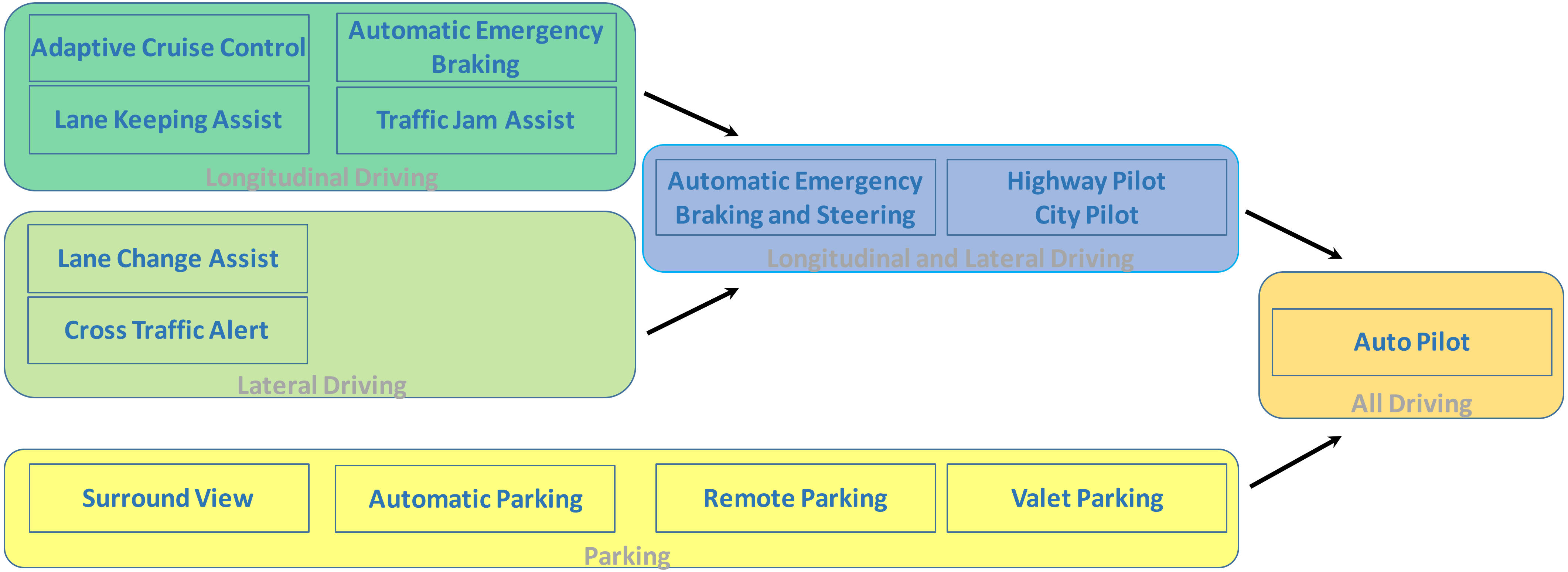


Self Parking

Autonomous Car

Driverless or robotic car capable of sensing its environment and navigating without human input

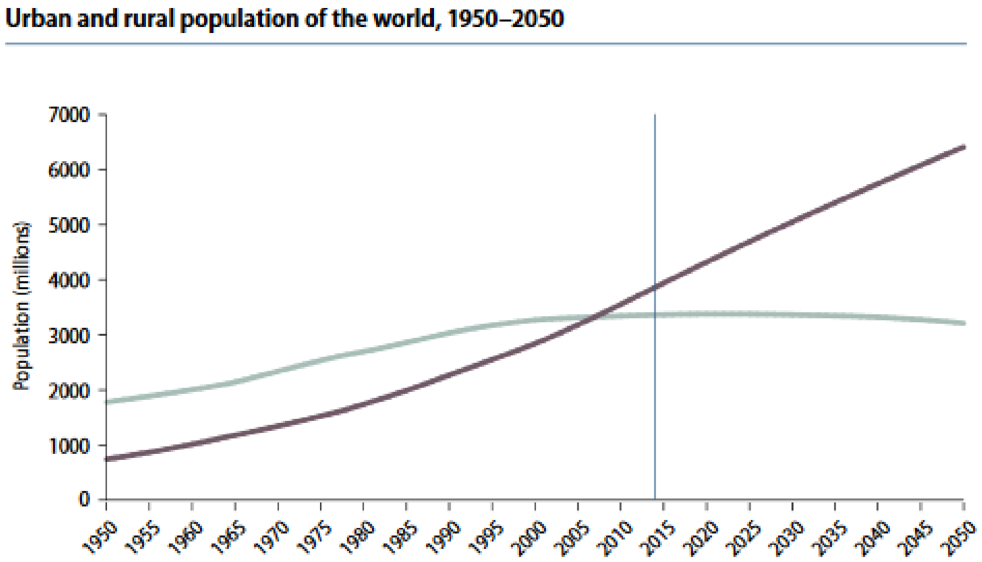
SAE Levels of Autonomy



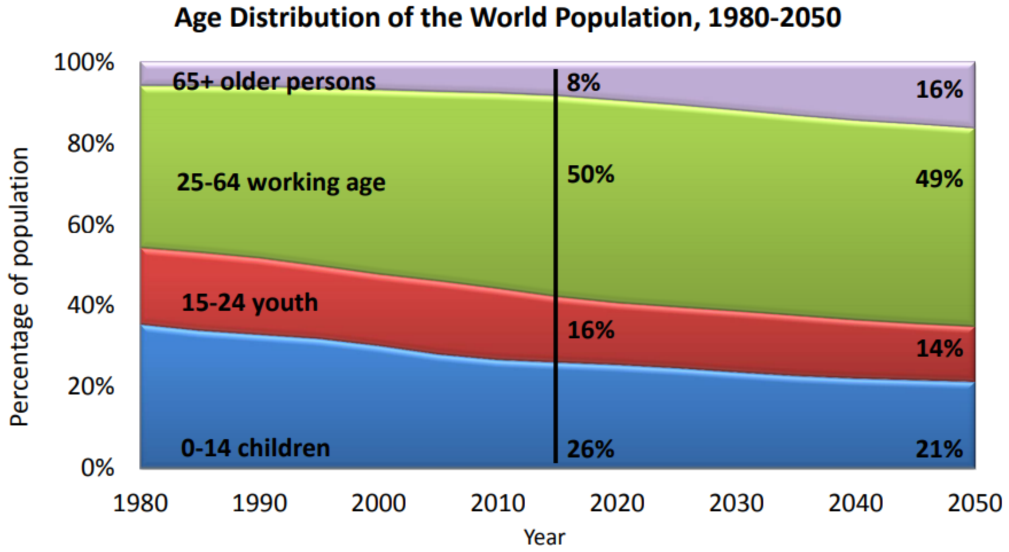
Level 0 No Automation	Level 1 Driver Assistance	Level 2 Partial Automation	Level 3 Conditional Automation	Level 4 High Automation	Level 5 Full Automation
a human driver has control of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems	a driver assistance system controls either a steering or acceleration/deceleration task with the human driver performing all remaining aspects of the dynamic driving task	a driver assistance system controls one or more steering and acceleration/deceleration tasks with the human driver performing all remaining aspects of the dynamic driving task	an automated driving system controls all aspects of the dynamic driving task with the human driver intervening when requested	an automated driving system controls all aspects of the dynamic driving task in specific scenarios	an automated driving system of all aspects of the dynamic driving task in all scenarios and environmental conditions

Why Autonomy?

- Societal benefits
  - Safety
    - Over 90% of accidents due to driver error
    - UK: 5 deaths per day, India: 400 deaths per day
  - Reuse road and parking congestion, air quality, parking
- Urbanisation



- Demographics
  - Car ownership - mobility as a Service
  - More non-drivers e.g. elderly



Sources:
UK World Urbanisation Prospects 2014
INRIX/Centre for Economics and Business Research Hiriko

UK Autodrive Project

- Project objectives:
  - Demonstrate autonomous and connected vehicles (V2X) in real-world urban environments
  - Provide insight for stakeholders including legislators, insurers and investors
- Vehicles:
  - Pods (RDM) in Milton Keynes
  - Autonomous (TMETC and JLR)
  - Connected (TMETC, JLR and Ford)

- Duration
  - 3 Years: November 2015 – October 2018
- Funding
  - Part funding from Innovate UK (around £10m of £19.2m)

Autonomous Functional Architecture

Sensing

Gather data from surroundings

Perception

(and localisation)

Interpret sensed data

Planning

Decide what to do

Control

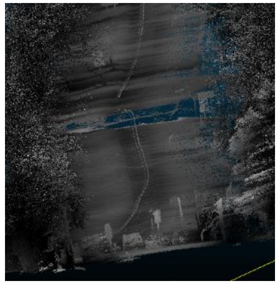
Achieve Desired Trajectory

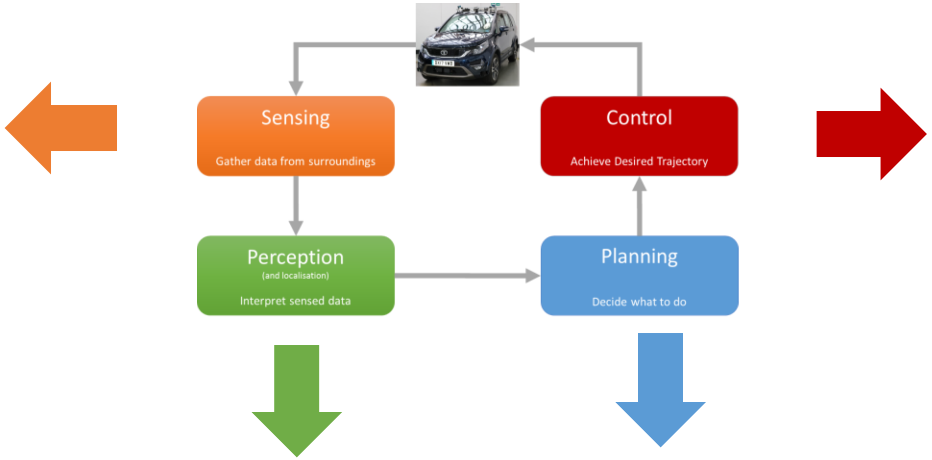
Linux
C++
Python

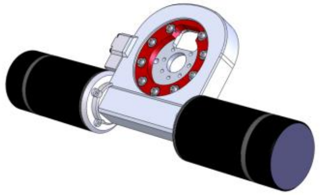
Windows
MATLAB
Simulink

ROS/PTP/CAN

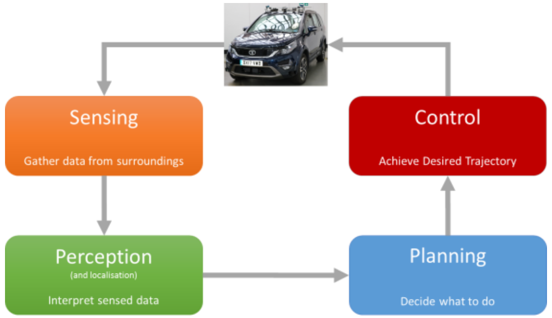
Autonomous Hardware Architecture

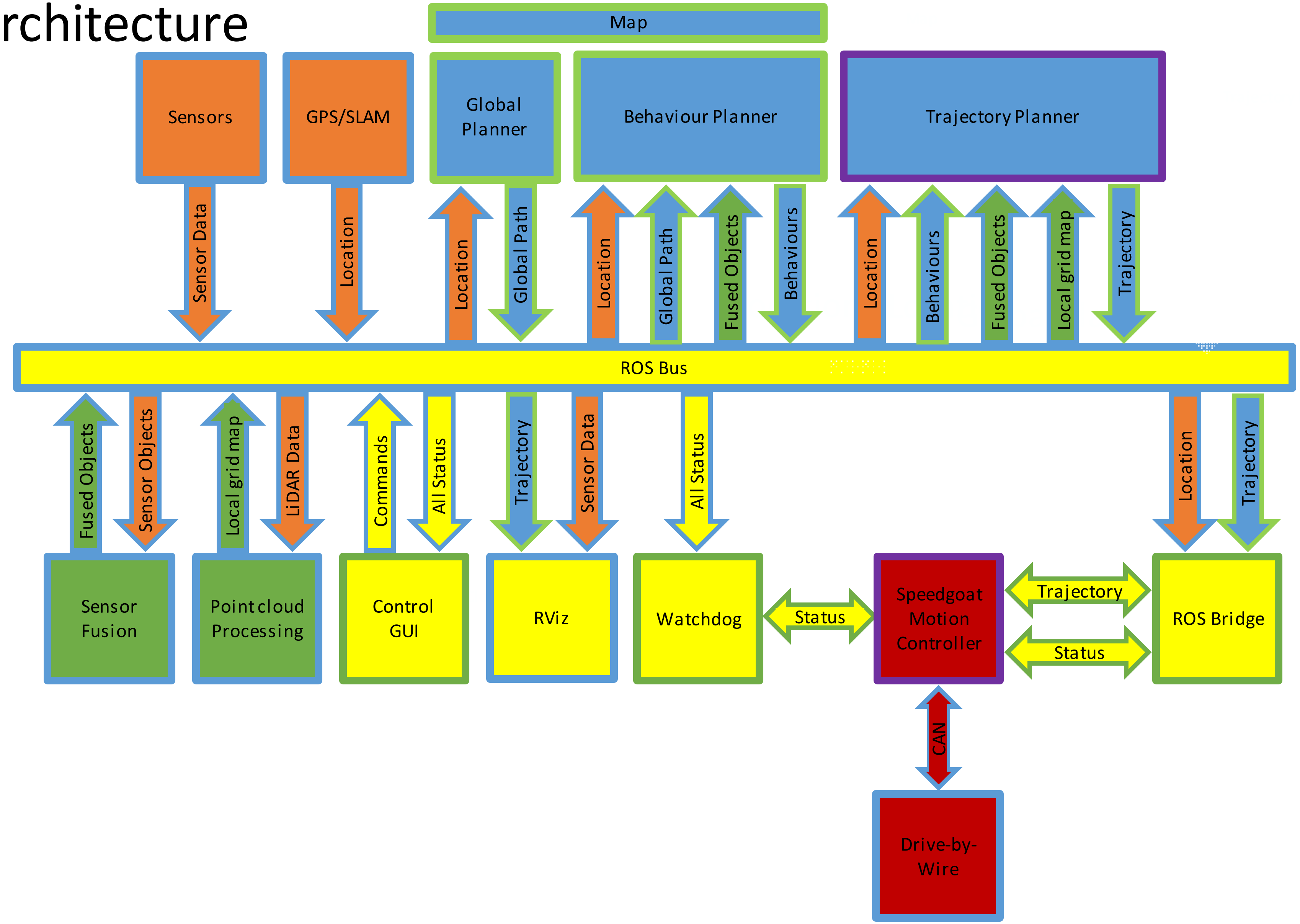






Autonomous Software Architecture





System

Supplier

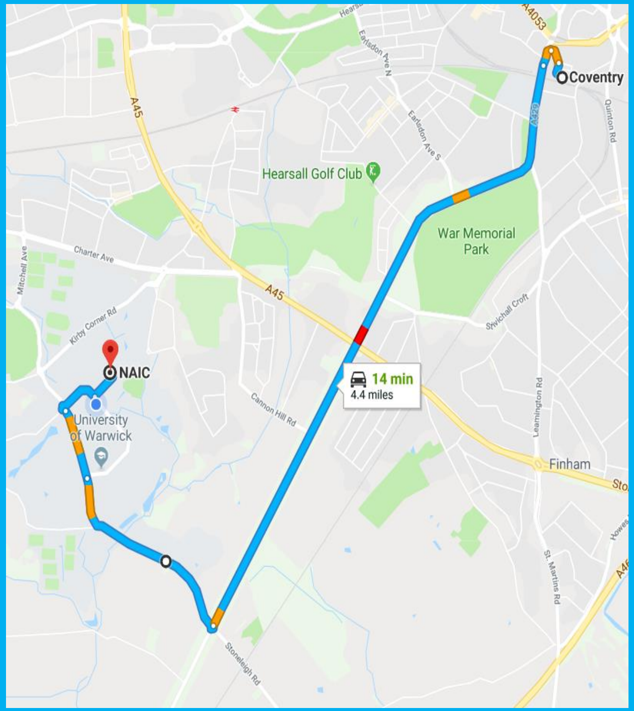
Linux/C++/Python

MATLAB/Simulink

Planning

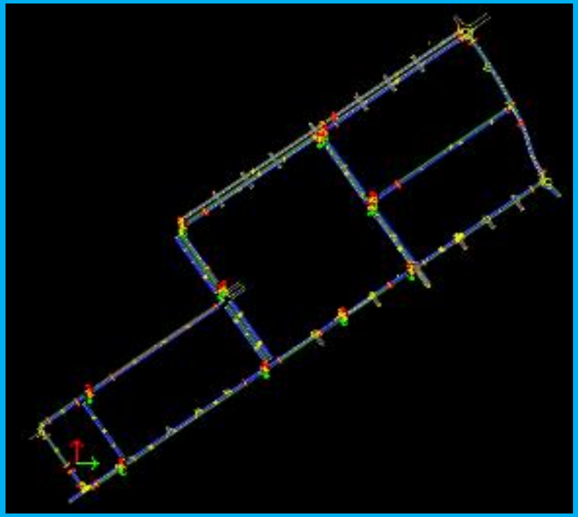
Global Planner

- Finds the optimal route to the destination (according to some time/cost objective)



Behaviour Planner

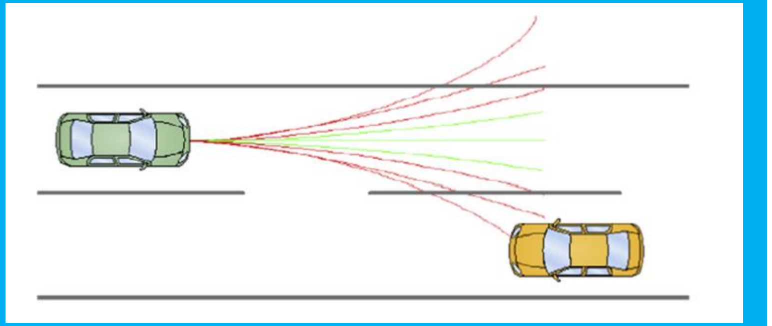
- Static (strategic) behaviours from the map e.g. keep in lane, stop, give way, left turn



- Dynamic (tactical) behaviours in response to the environment e.g. traffic lights and objects (and evasive trajectories to mitigate risk)

Trajectory Planner

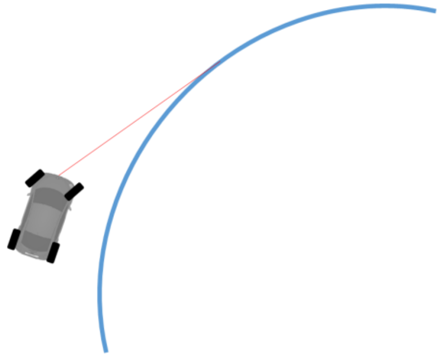
- Generate obstacle free paths
- Assign speed profiles



- Select trajectory
  - within lane boundaries, avoiding dynamic/stationary obstacles, comfortable (yaw rate; lateral and longitudinal acceleration and jerk; meets regulatory constraints (speed limits, stop lines, traffic lights etc.)
- Select evasive trajectory

Control Options

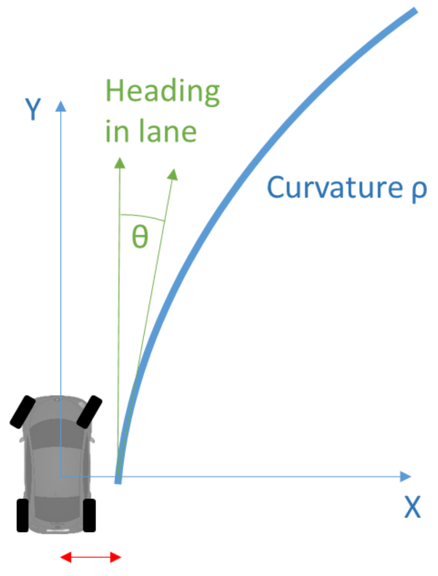
- Pure Pursuit
  - Look ahead distance sets path intersection point
  - Steer wheels to match angle to intersection point
  - Accuracy/stability dependent on look ahead distance (include yaw damping)



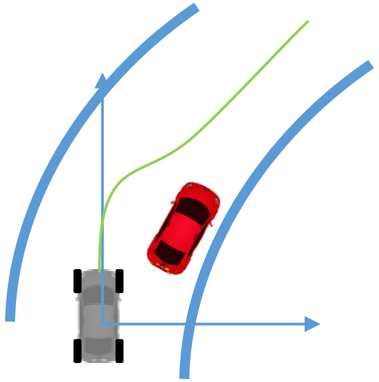
- Lane Keeping
  - A quadratic equation approximates the path to be followed by the vehicle

$$X = xo + \theta Y + \frac{\rho}{2} Y^2$$

  - Controller has three loops Position/Heading/Yaw rate loops speed scheduled gains
  - For lower speed higher curvature turns it was considered that accuracy would not be sufficient
  - Vehicle converges to an ‘ideal path’ so the ideal path and converging path need to be obstacle free

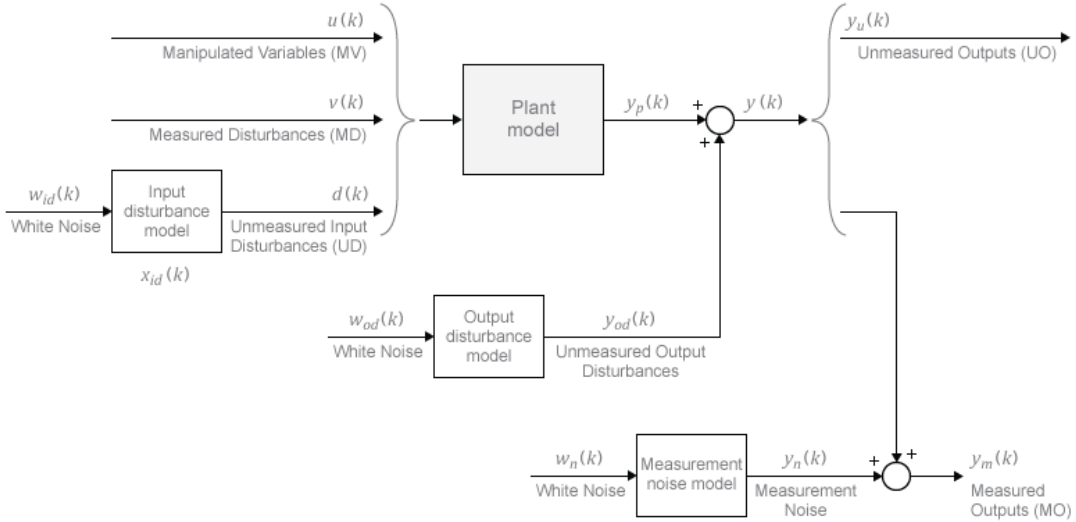


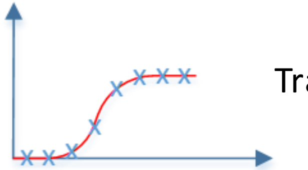
- Trajectory Tracking
  - More intuitive path creation starting at current position
  - Track a trajectory that has been already created to avoid of obstacles



Control Method

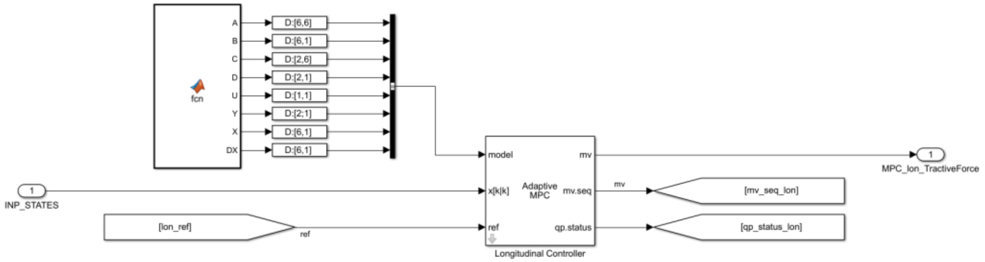
- Model Predictive Control
  - Inputs
    - Trajectory for a given time ahead
    - Dynamic model of the vehicle

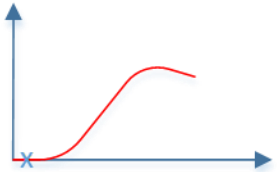




Trajectory

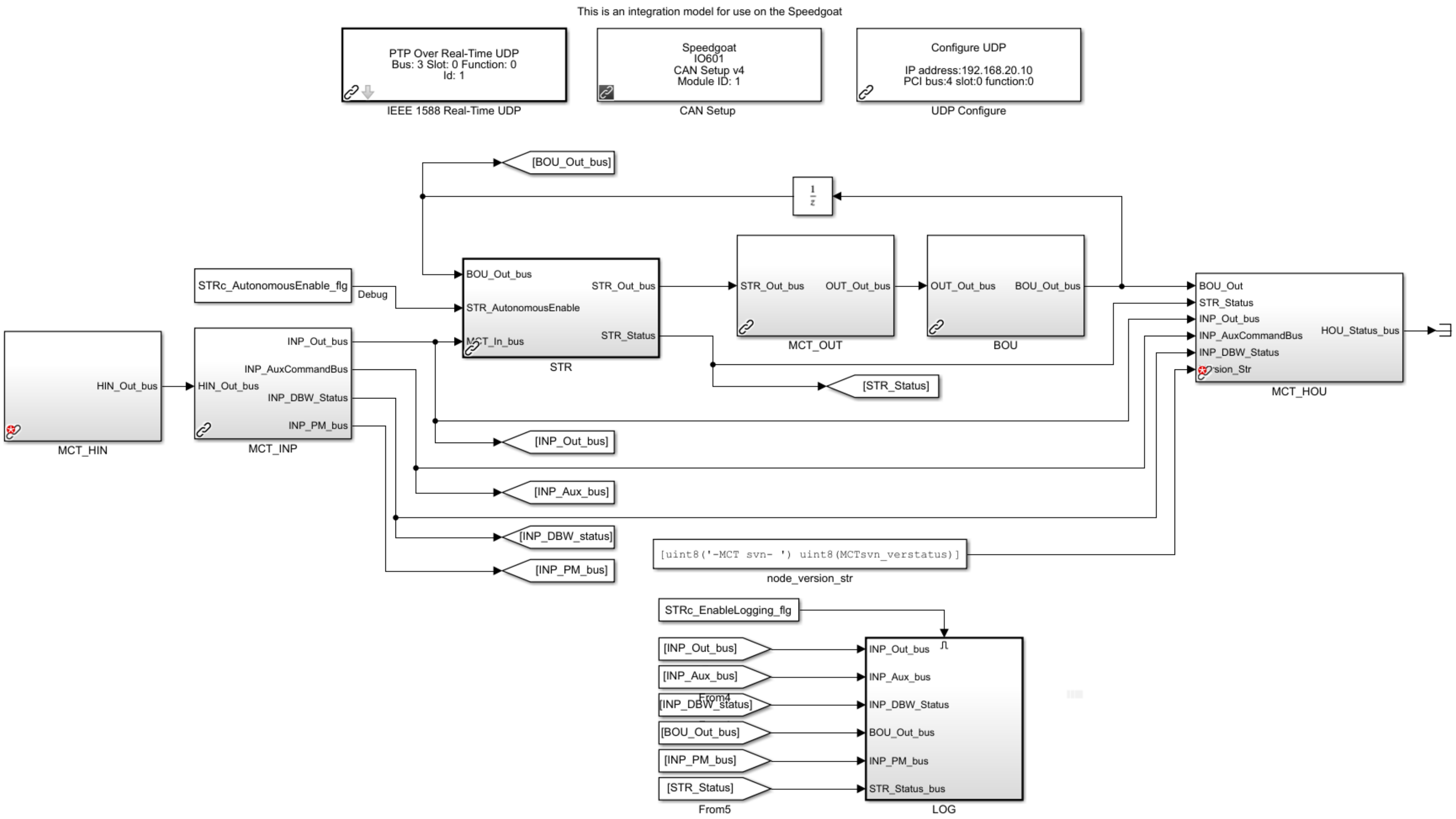
  - Outputs
    - Vehicle control sequence needed to track the trajectory by minimising a cost function for the given mode





Control

Control Implementation - Simulink

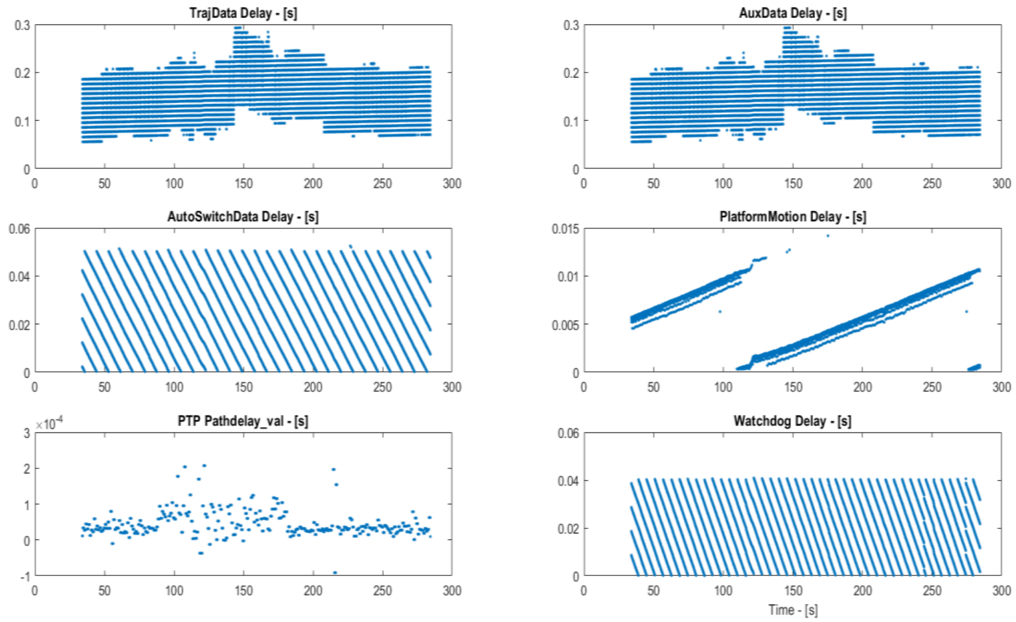


Control Method

- Model Predictive Control Design Considerations
  - Define ‘prediction horizon’ – how far in time you are looking ahead
  - Define ‘control horizon’ – how far in time you are predicting control for (control is then fixed for remainder of prediction horizon)
  - Optimal control sequence that could follow the trajectory is generated but all the control terms are discarded except the first one
  - Models:
    - MPC embedded
      - Model fidelity traded off against available online computational resources
      - Zero speed needs consideration (dynamic bicycle model is singular at zero speed)
    - Closed loop offline simulation
      - Complexity/fidelity versus simulation time
  - Delays - old trajectories/old measurements as distributed processing and data transfer time delays (not fixed as non-real time network)
  - Measurement reference and conventions e.g. centre of gravity or rear axle
  - Linear analysis is complicated with non-linear components (particularly trajectories i.e. Bèzier curves) and time delays in the loop so guaranteeing stability control performance analytically is difficult

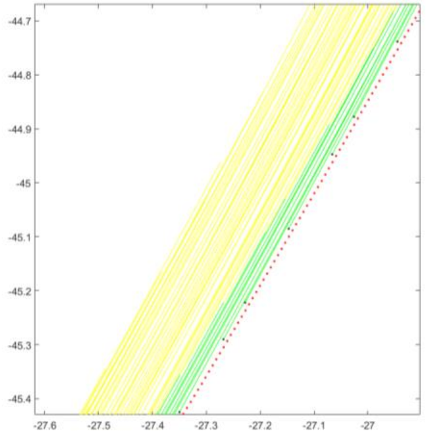
Time Issues

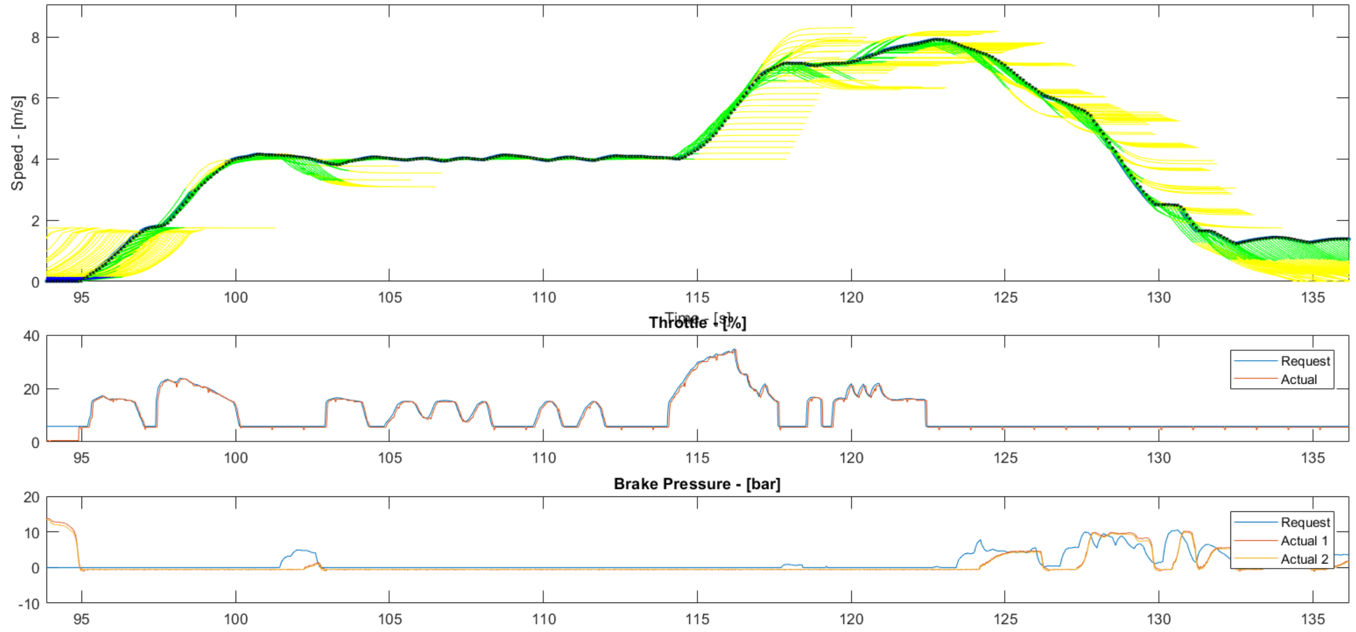
- Challenge
  - Data ages from the disparate sources is needed
    - It is critical to use current up to date information
    - Delayed data needs to compensated for (and ideally the delays minimised) as
      - control loops are notoriously harder to stabilise and give accurate responses (larger errors and or lagged responses) when using delayed data
    - Ages needed to reduce errors in projecting states into the future (e.g. objects future position)
- Solution
  - Synchronise the different processors clocks - Precision Time Protocol (PTP)
  - Time stamp data
- Results
  - Relative age of data is known but the ages variable (but are bounded)



Data Logging and Visualisation

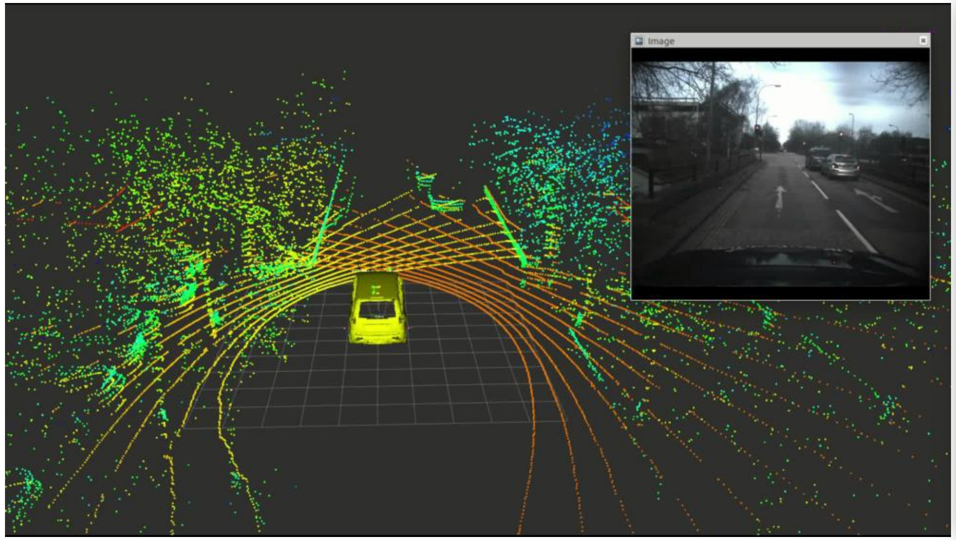
- Requirements
  - All on-road data is logged - e.g. UK Department of Transport ‘The Pathway to Driverless Cars: A Code of Practice for Testing’
  - Playback visualisation for analysis/debugging
- Solutions
  - ROS and Speedgoat logging
  - Bespoke offline visualisation

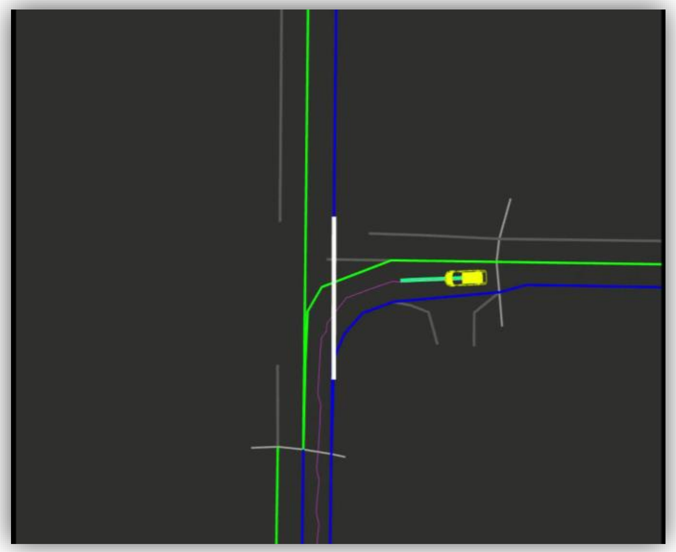


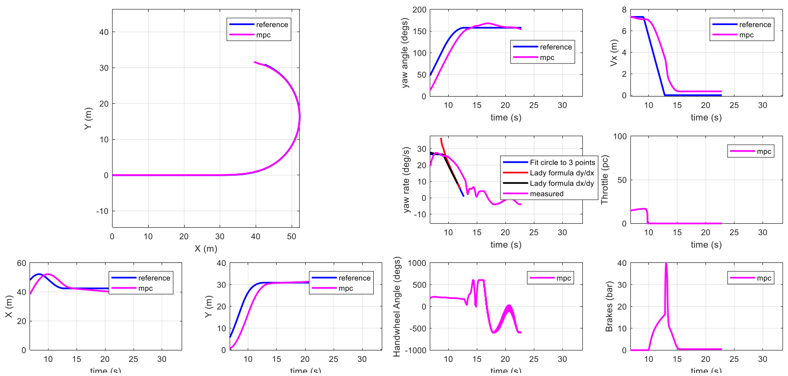


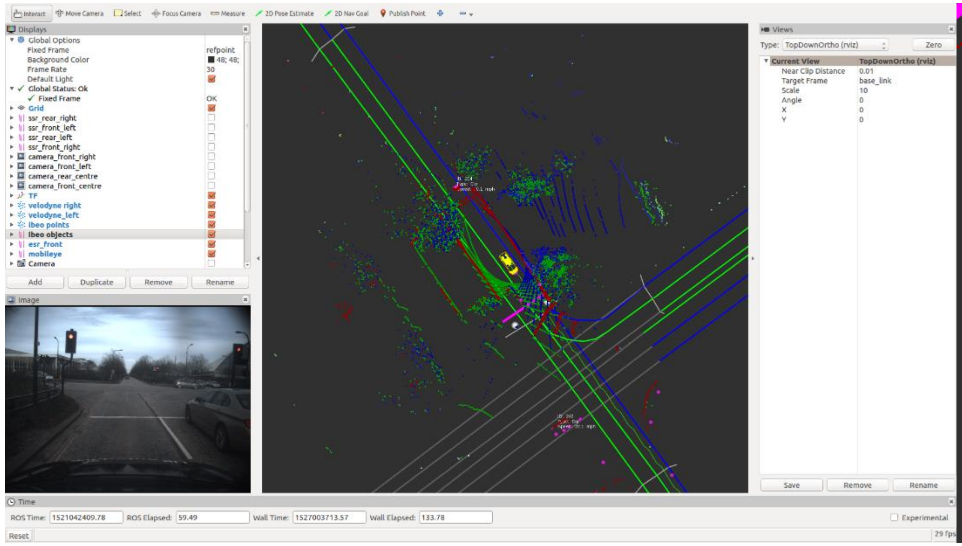
Data Logging and Visualisation

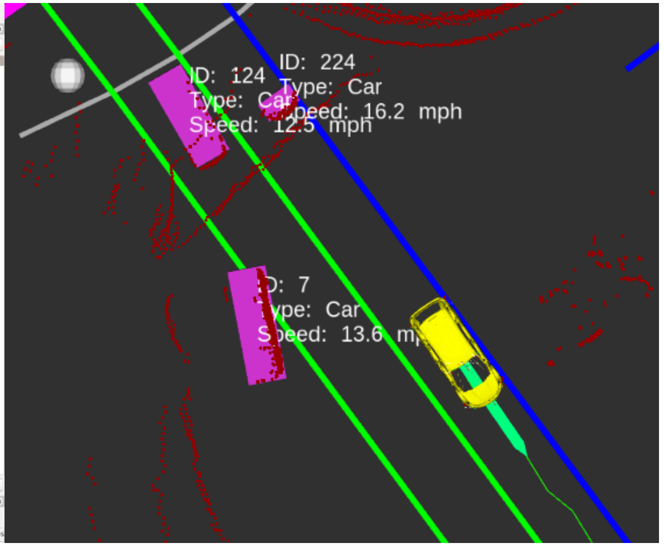
- Enhancements
  - Online visualisation and logging capability



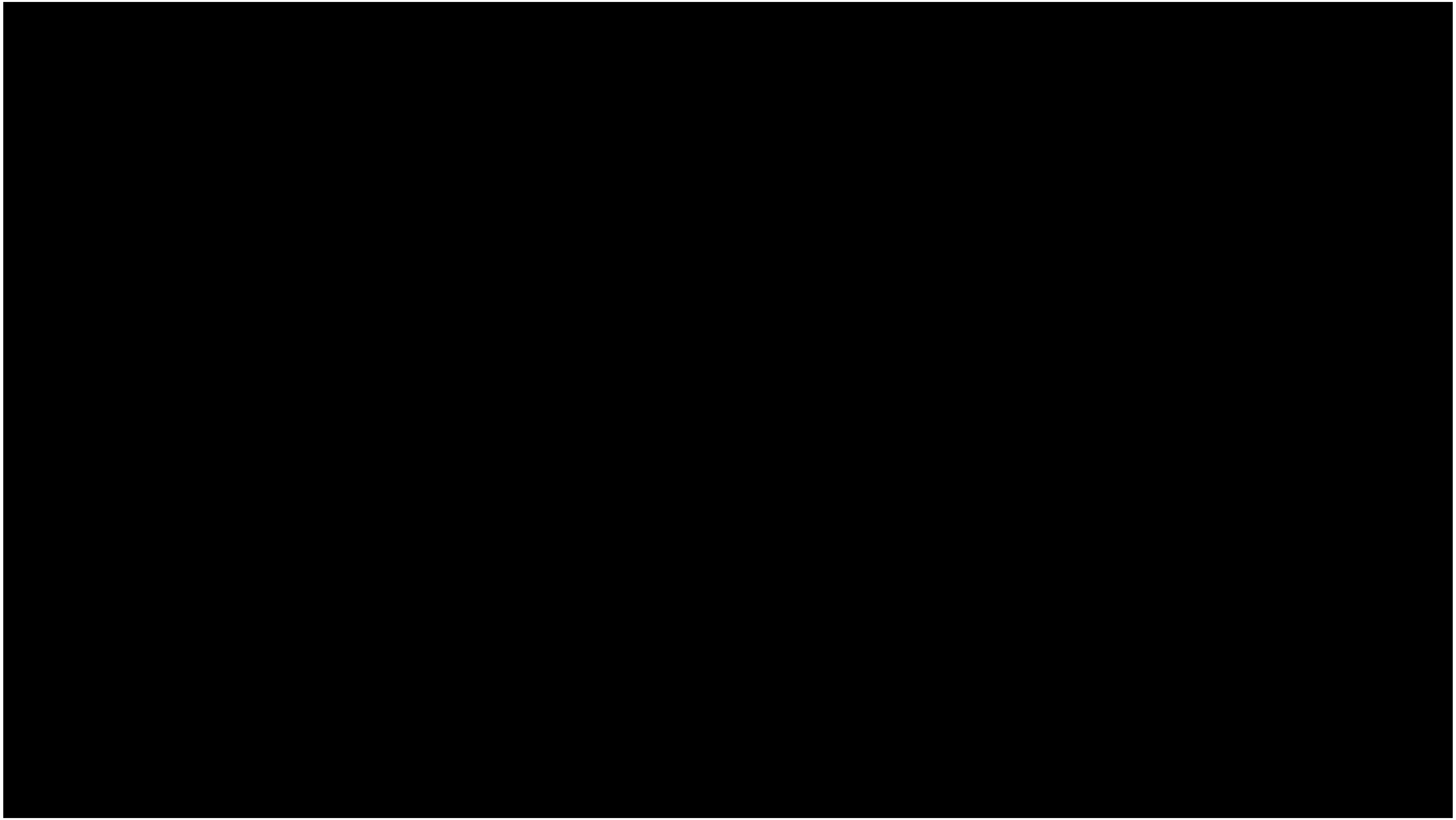








Autonomous Hexa During Coventry and Milton Keynes Trials



Conclusion

- Pragmatic approach to Autonomous Vehicle Development
  - Off-the-shelf Tools
    - Linux/C++/Python/RViz/CANalyzer/MATLAB/Simulink
  - Off-the shelf hardware
    - Radar/LiDAR/GPS/IMU/cameras
    - Industrial PCs/Speedgoat/Drive-by-wire mobility solution
  - Bespoke third party software
    - Speedgoat bridge
    - Sensor Fusion
  - TMETC Software
    - Perception/planning/motion control
    - Sensor fusion
- Future Work
  - Third party and in-house tool enhancements
  - Lessons learnt to improve the system and develop it further towards higher autonomy levels
- Undergoing UK Autodrive Trials, Demonstrations and Dissemination in Coventry and Milton Keynes now (October 2018)

Finally

- Thank you
  - TMETC - Maradona Rodrigues , Eliot Dixon, Lorenza Gianotta, Andy Harris, Johnathan Breddy, Jon Clark
  - MathWorks – GianCarlo Pacitti

- Any questions?