



Drone assisted truck navigation

using model based development

Karel Kural & Jan Benders
HAN Automotive Research

Jan.Benders@han.nl

Karel.Kural@han.nl

HAN
University of Applied Sciences

≈35.000 students
≈3.500 employees

Faculty of Business
and Management
(ABS)

Faculty of
Health and
Social Studies

*Faculty of
Engineering*

Faculty of
Education

Informatics and
Communication
Academy

Institute of
Engineering

*Institute of
Automotive
Engineering*

Institute of Built
Environment

Institute of Life
Sciences

*HAN
Automotive
Research*

Automotive
laboratory

Automotive
bachelor/master
education

i-Mobility

Green Mobility

Control
Systems

Powertrains

Vehicle
Dynamics

Structural
design

≈20 FTE practice oriented researchers
together with many students

Driver challenges while maneuvering/reversing

Limited field of view confirmed by eye tracking experiments

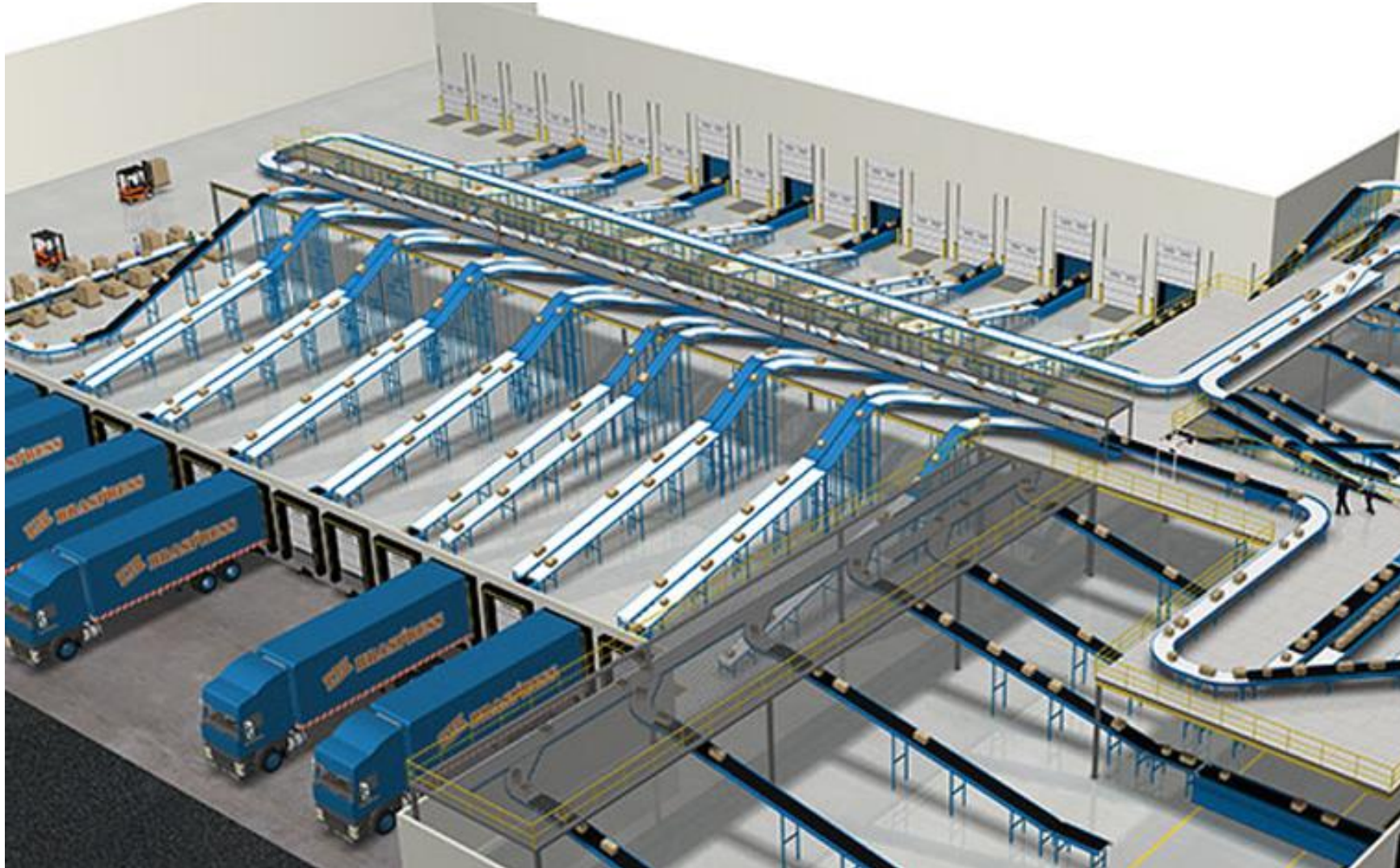


Divergent directional instability
when reversing

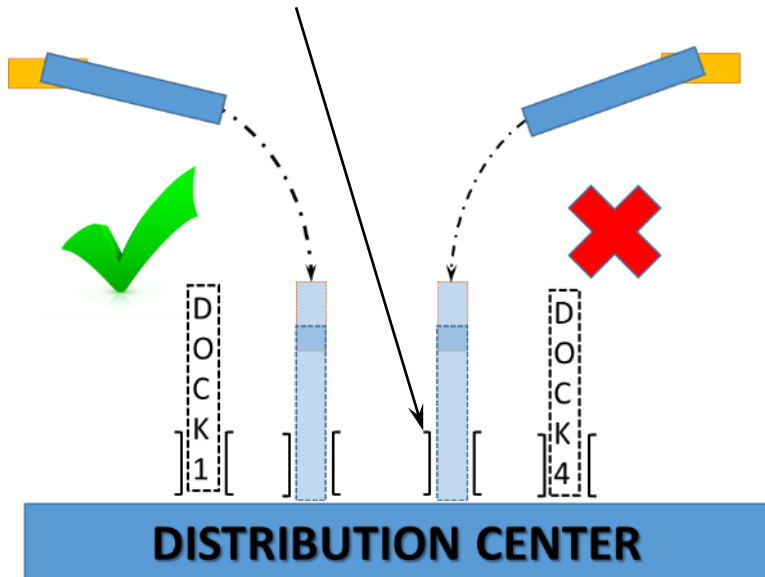
Each vehicle combination
behaves differently



Crucial maneuvering scenario: Docking

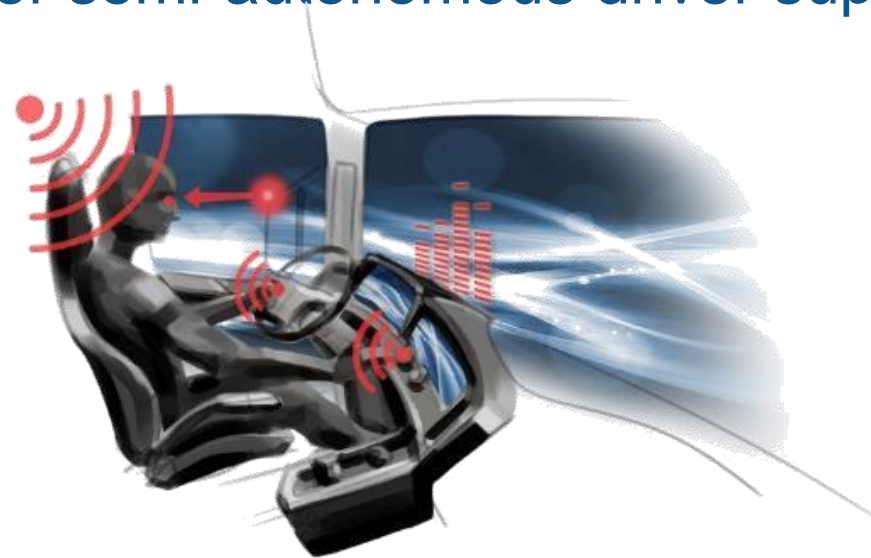


Docking: Challenges for the driver



Autonomous or semi-autonomous driver support systems

but....



...we need an accurate referencing system (GPS is not sufficient)

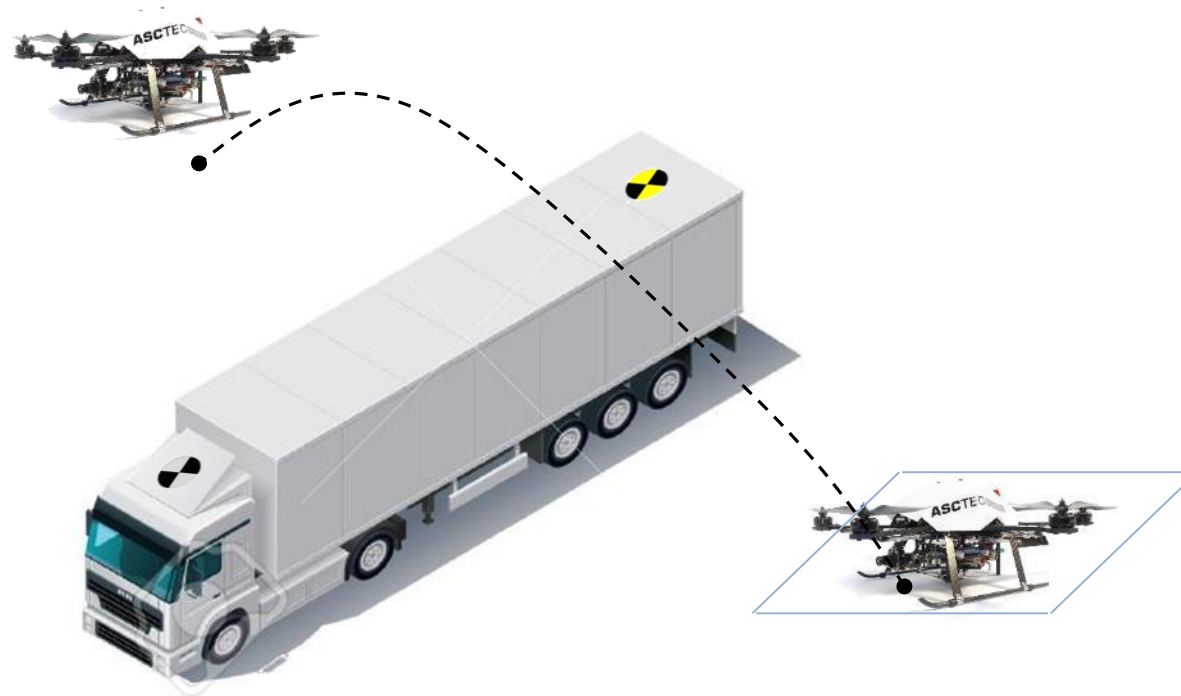
Some alternatives:

- Magnets/transponders in the pavement
- Laser scanners at docking gates
- Drone with camera up in the air

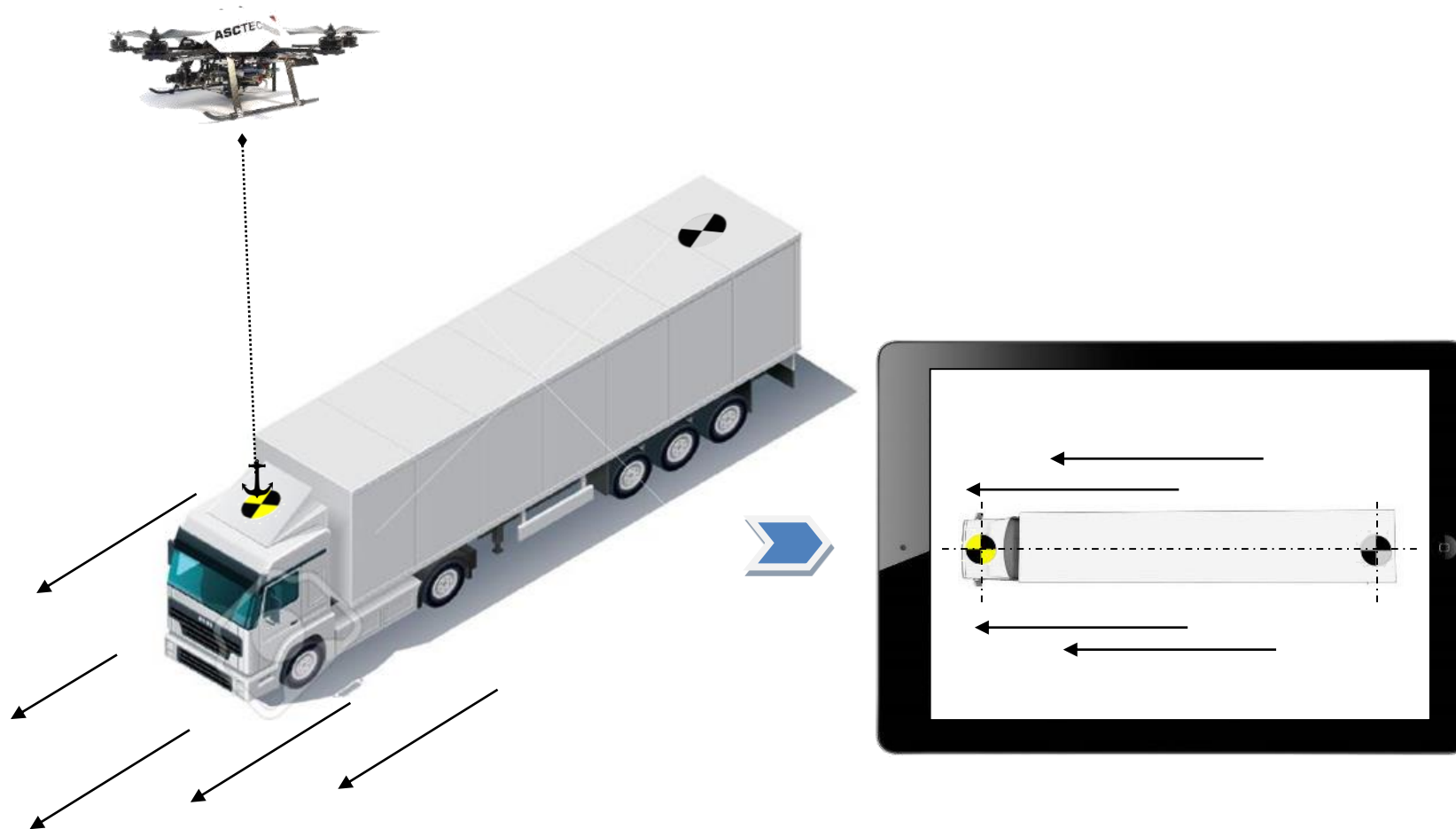
- Provide the driver with top view of the situation
- Continuously monitor the position of the vehicle w.r.t. destination
- ‘Last mile’ navigation with precise backwards docking support



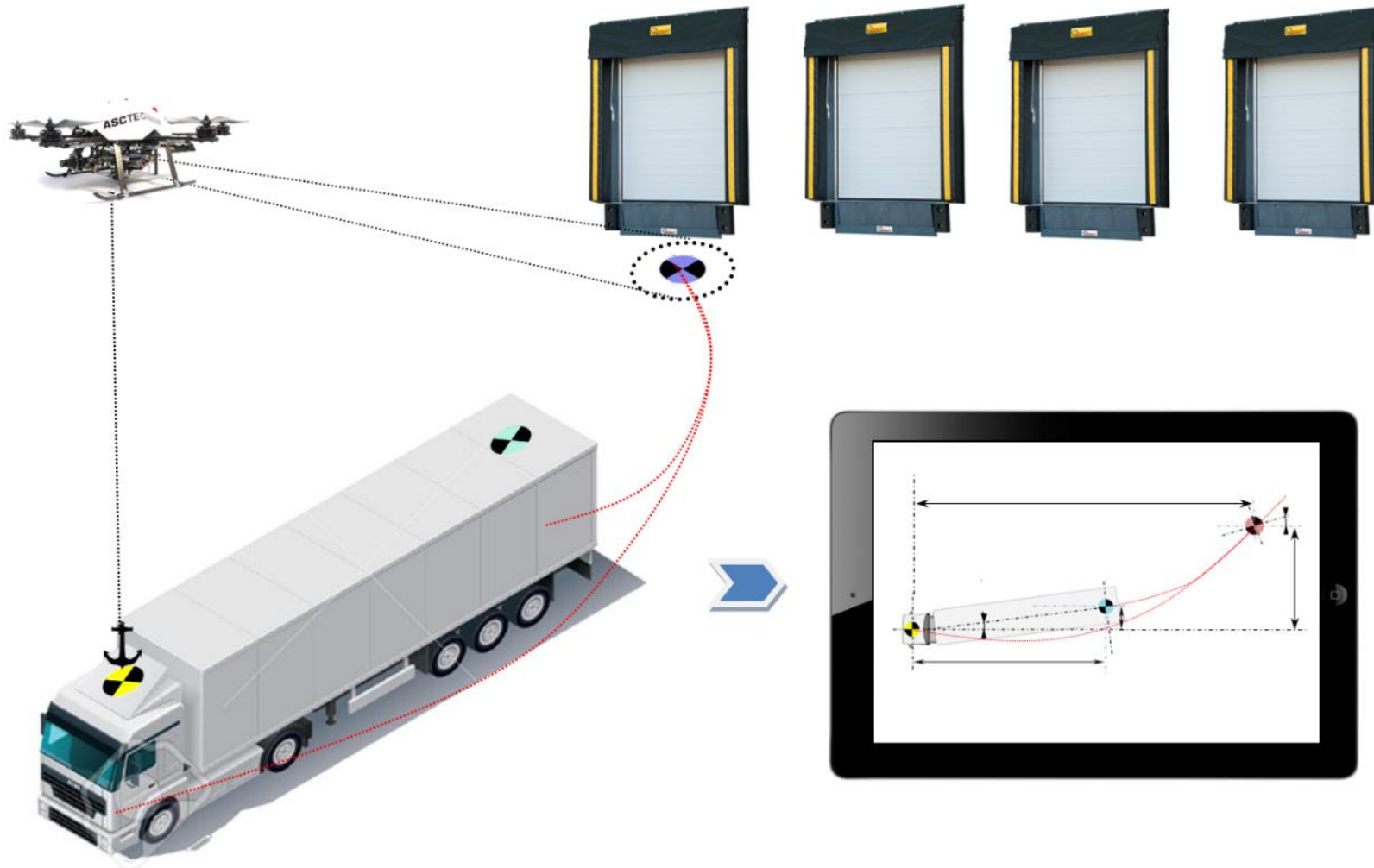
Find the vehicle combination nearby DC while connected to WMS



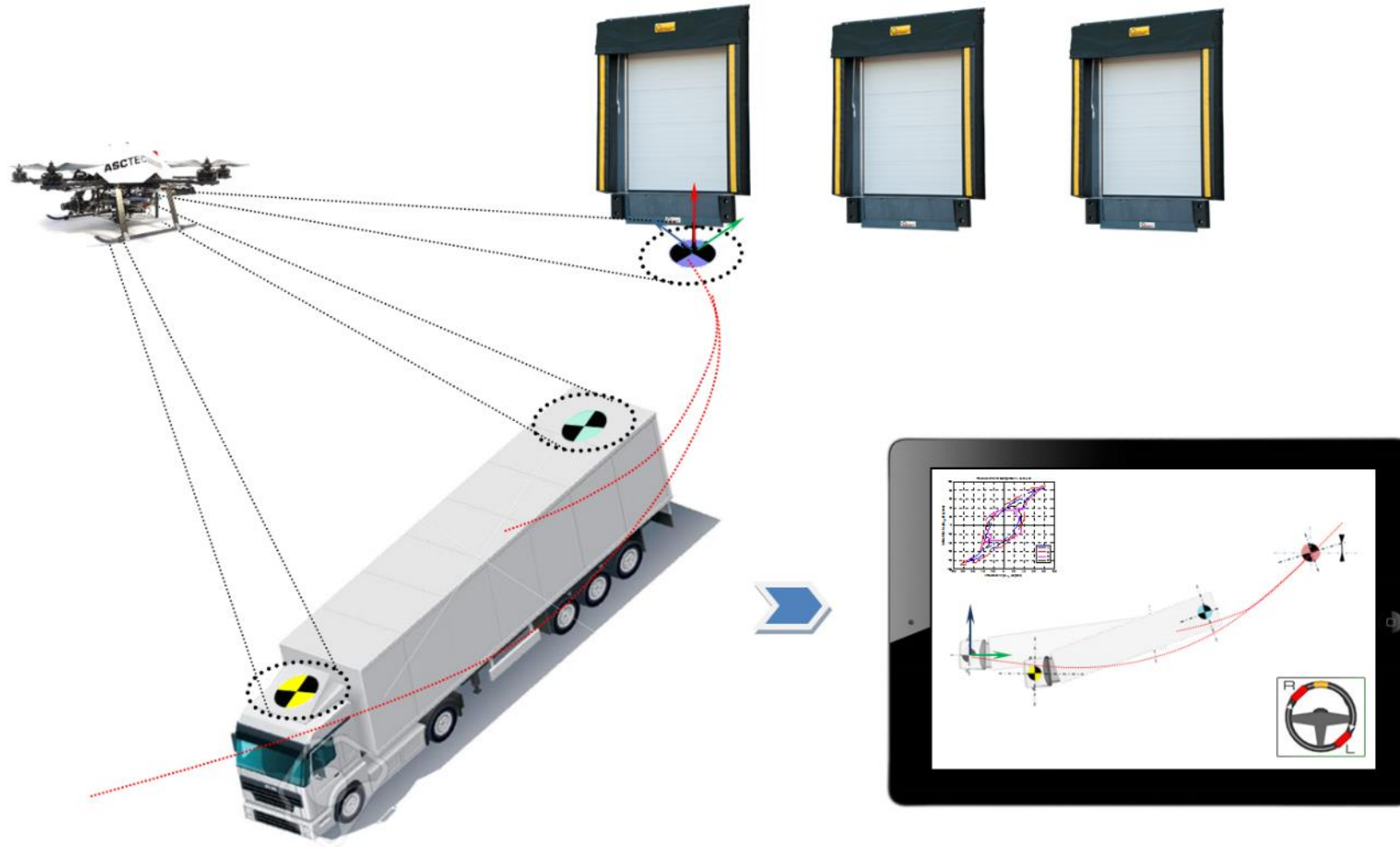
Anchor & navigate the truck nearby the gate (GPS)



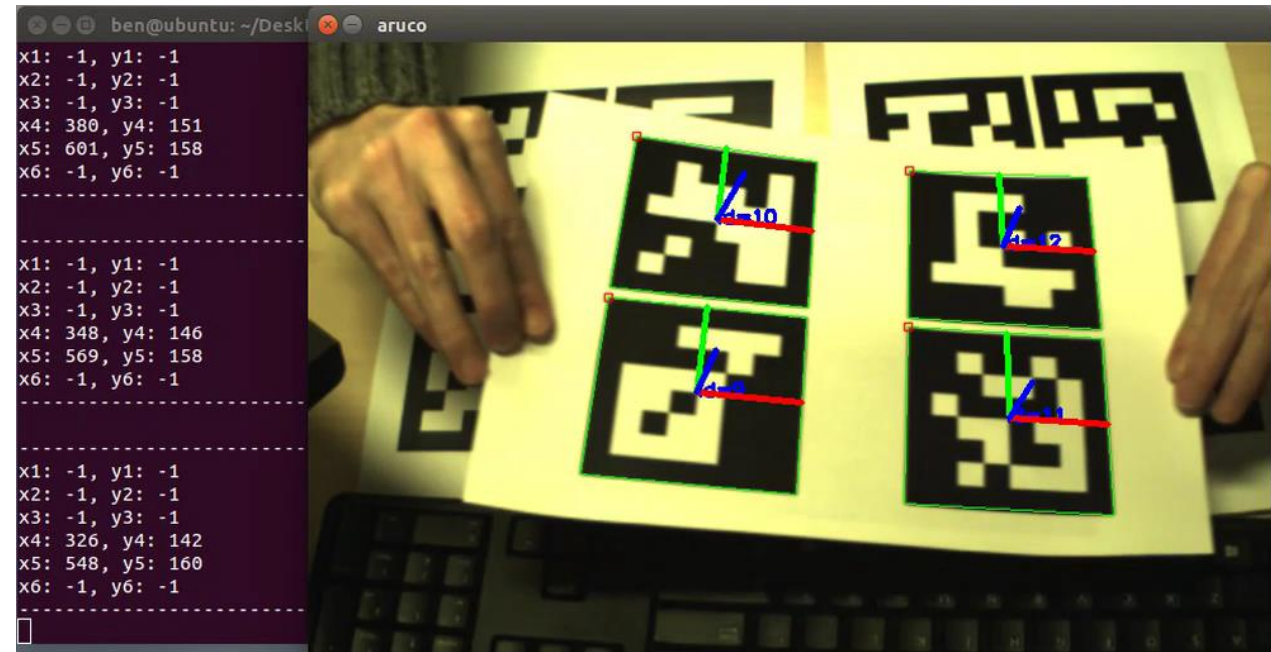
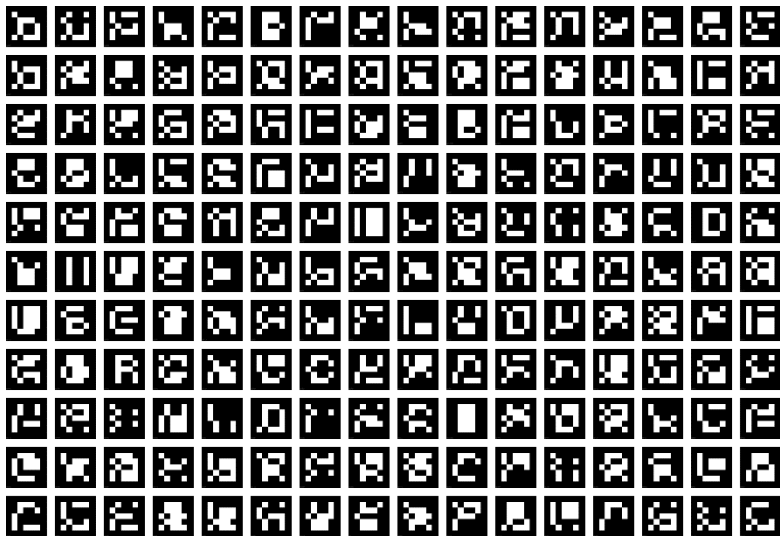
Detect the gate → Measure mutual position → Plan path



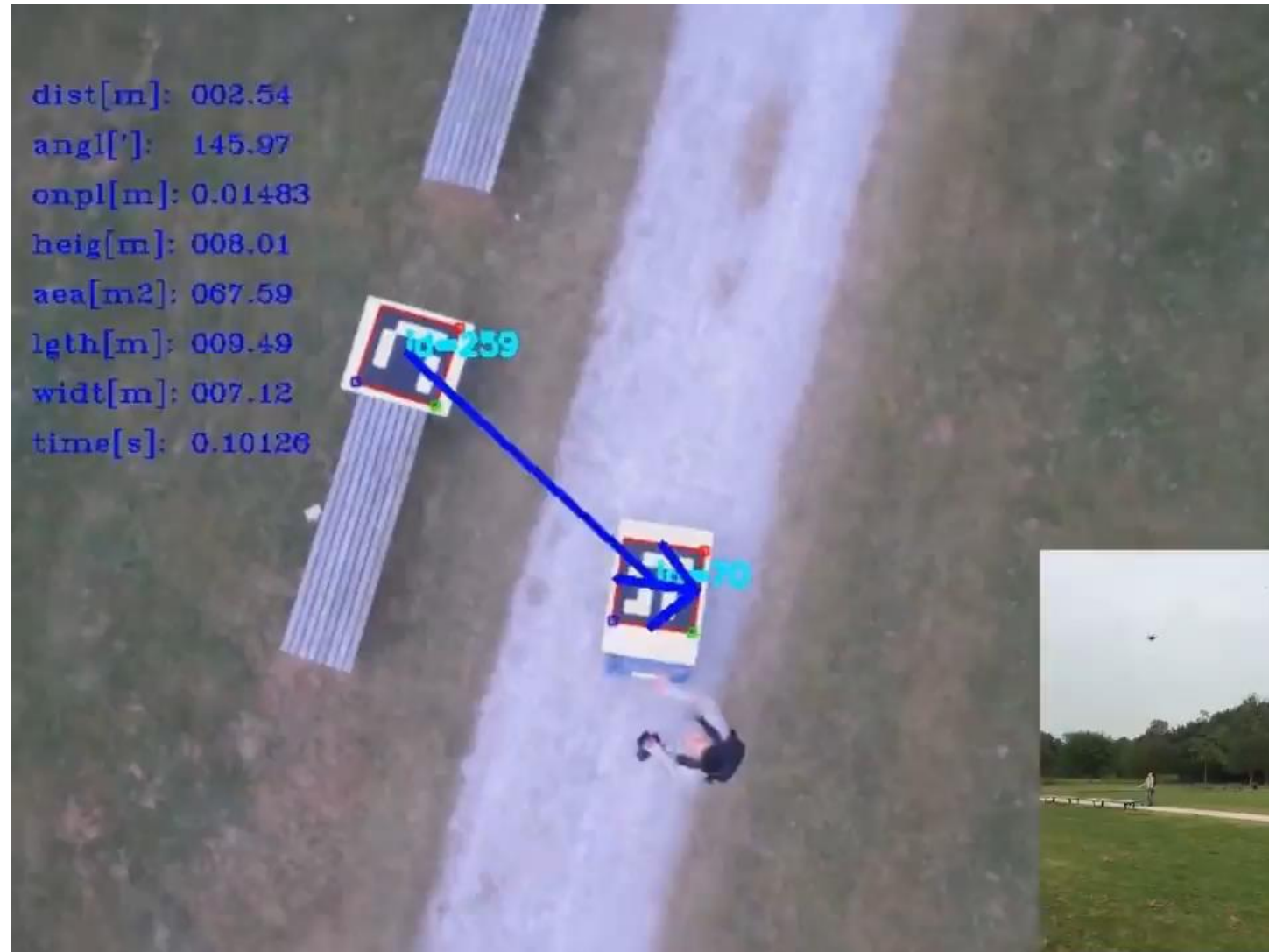
(Supported semi-) Autonomous path-following



- Using computer vision
- Temporary solution: ArUco* markers
 - Identification
 - Localization



Demo ArUco detection and following by drone



Demo of drone following a truck



- Planner role:

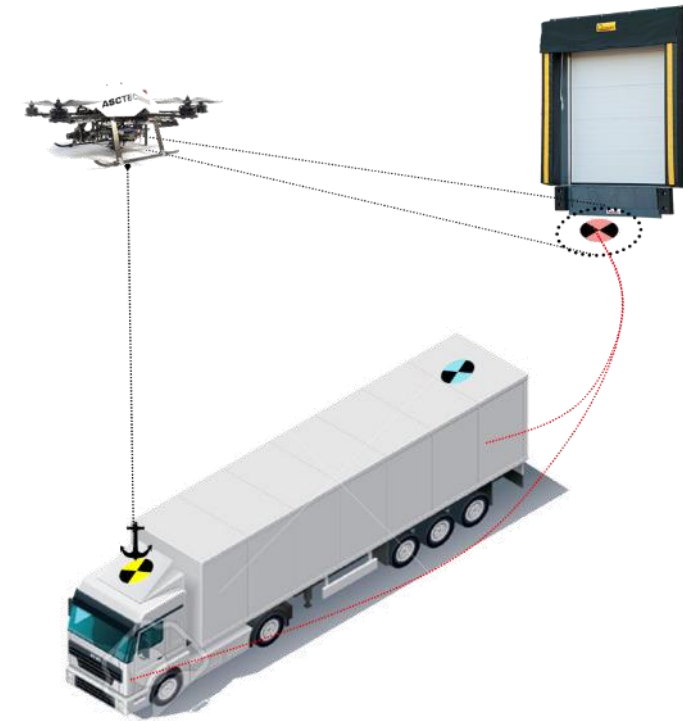
To generate a path between initial position and the destination that is:

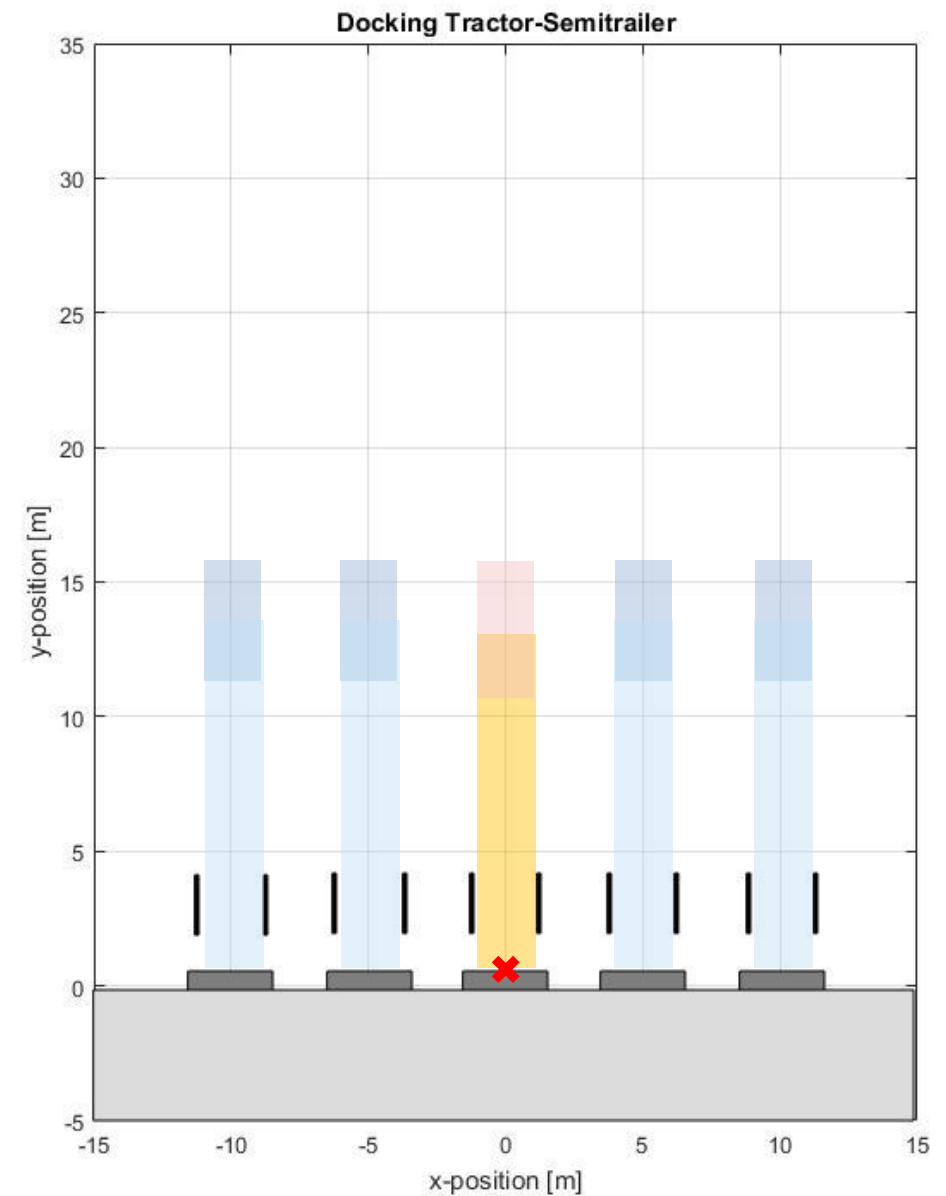
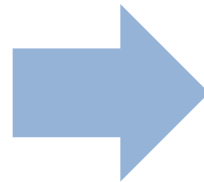
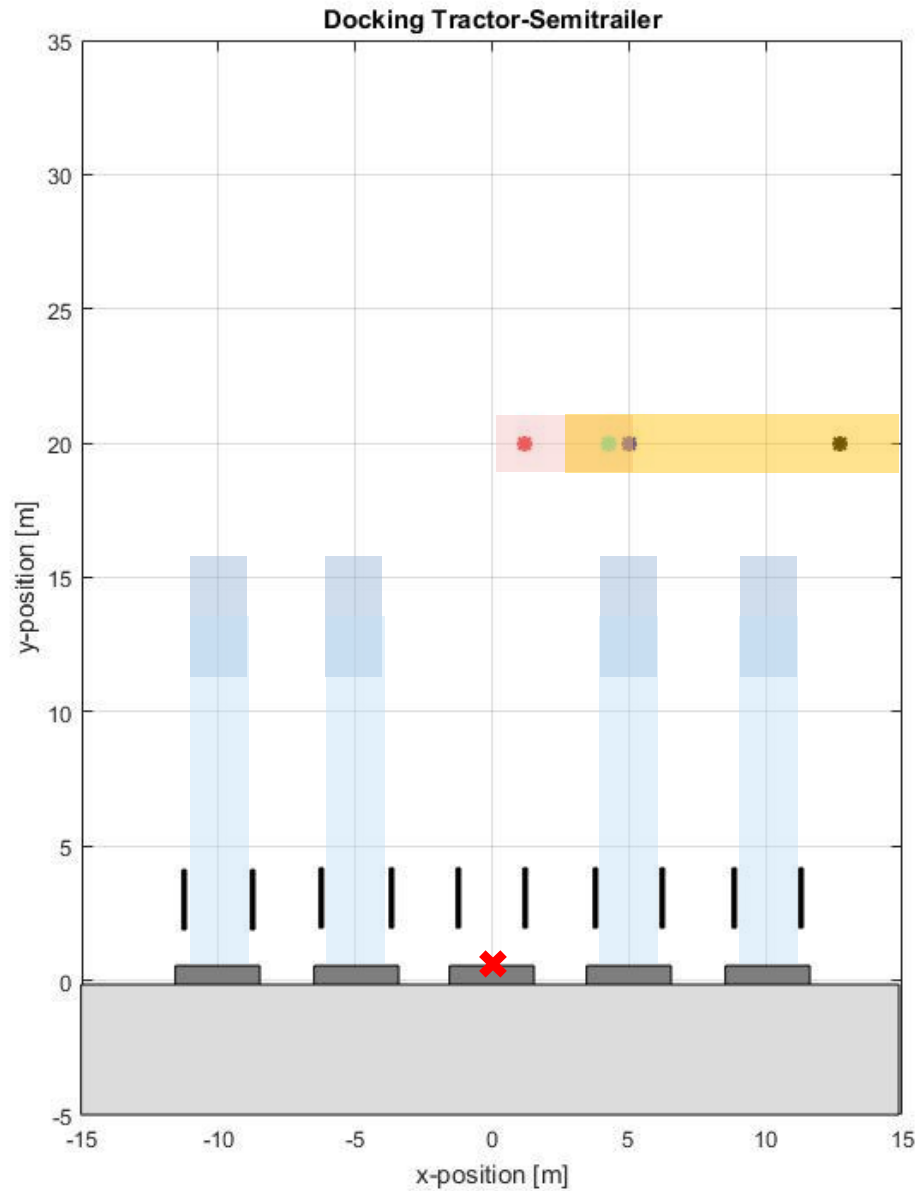
- Continuous
- Kinematically possible
- Efficient (minimize maneuvering space)

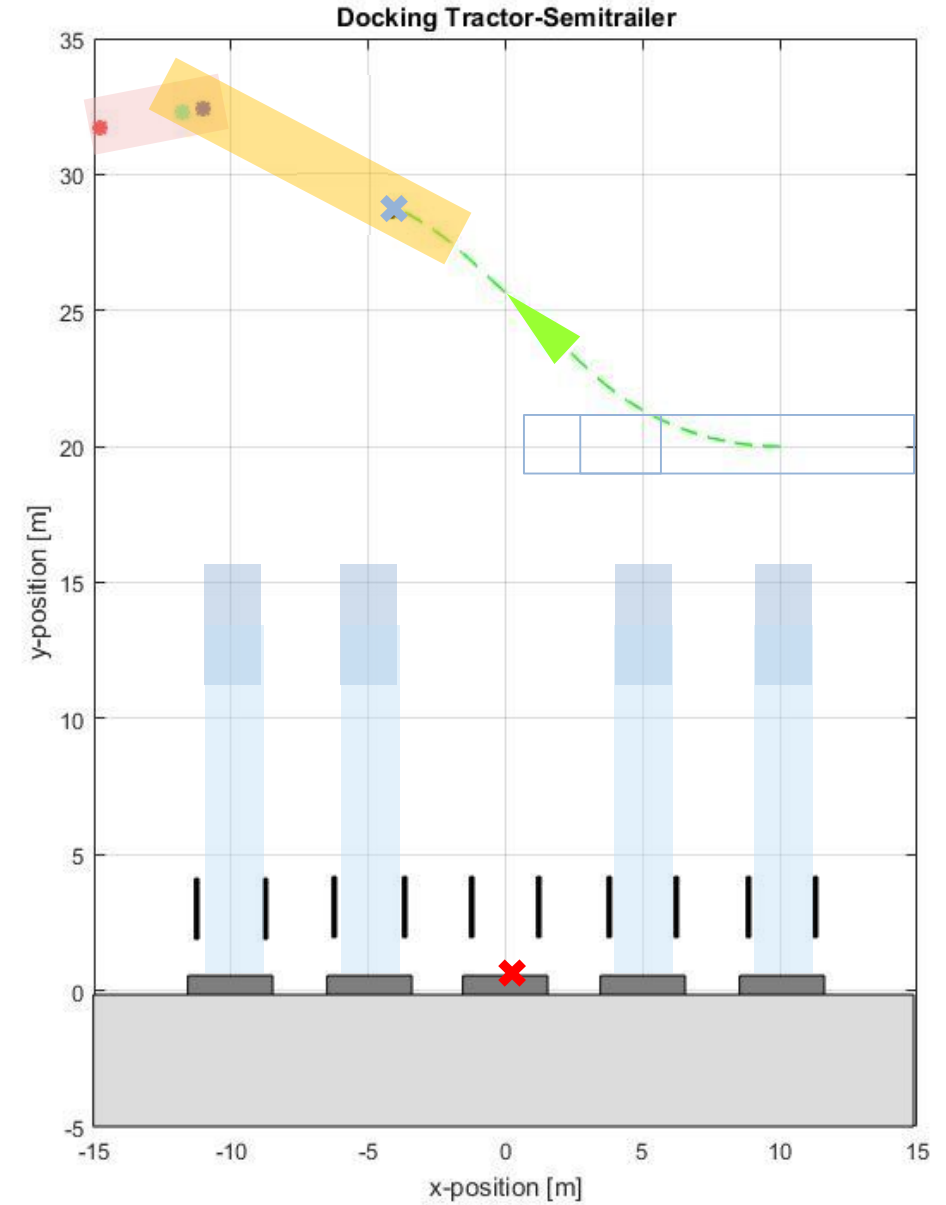
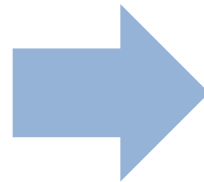
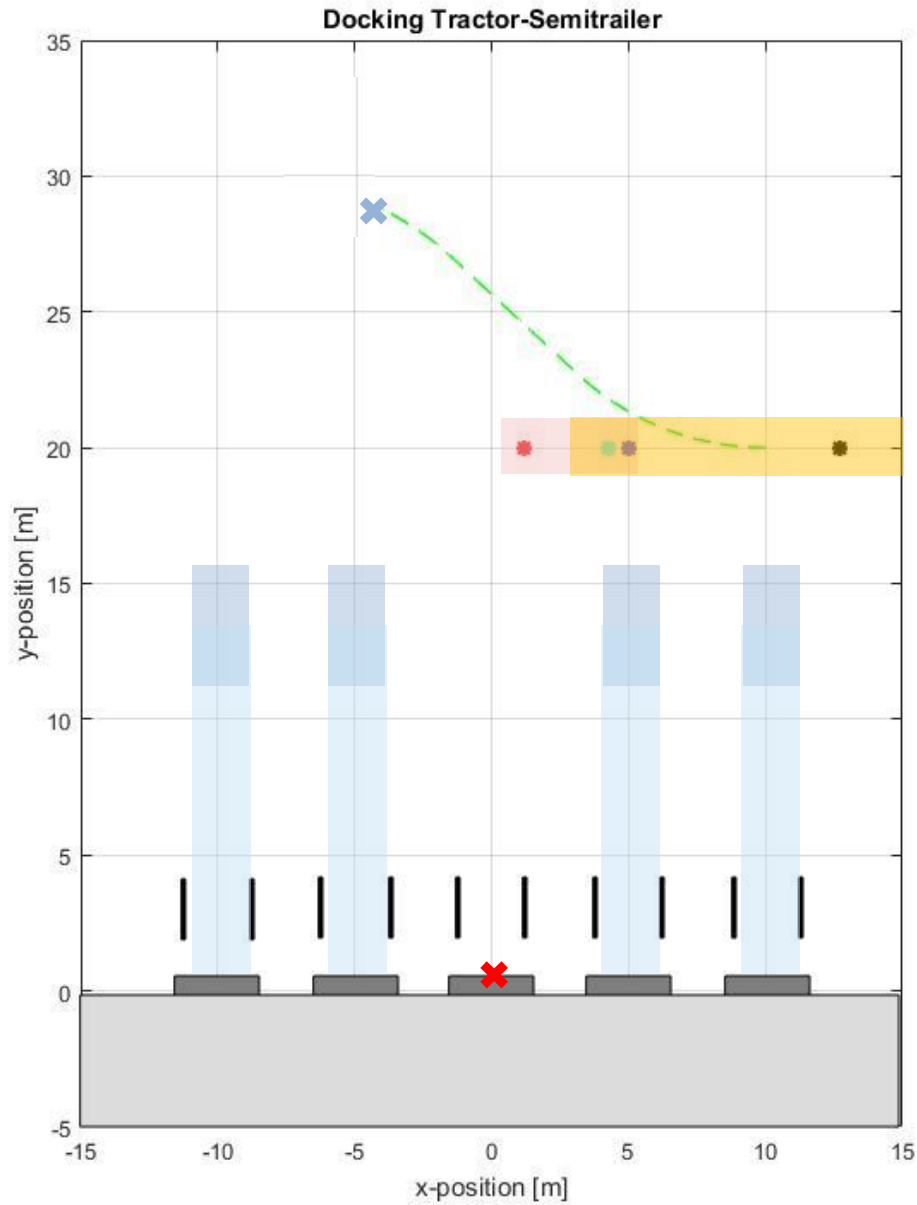
- Controller role:

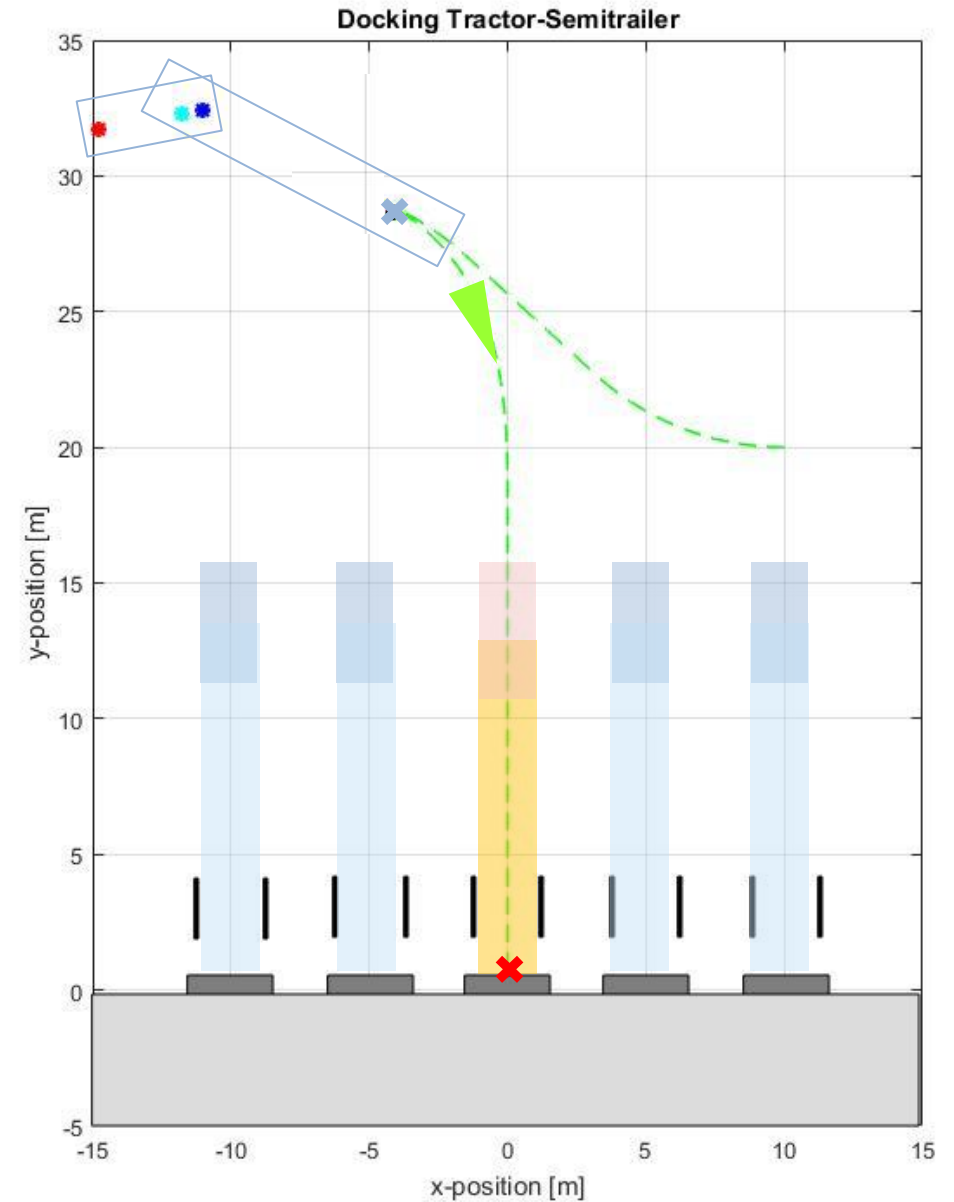
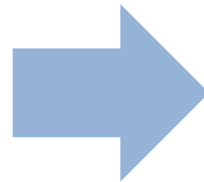
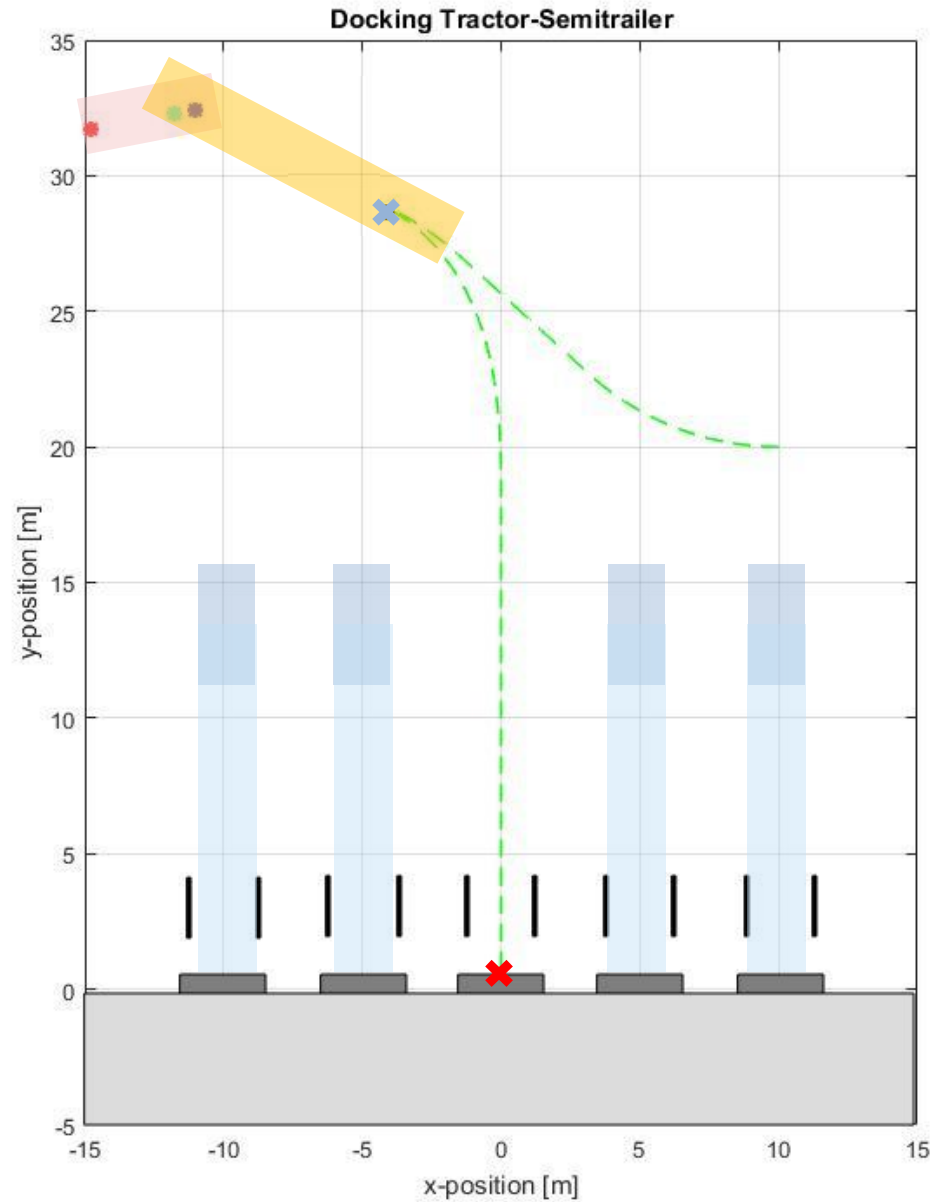
To navigate the vehicle over predefined path:

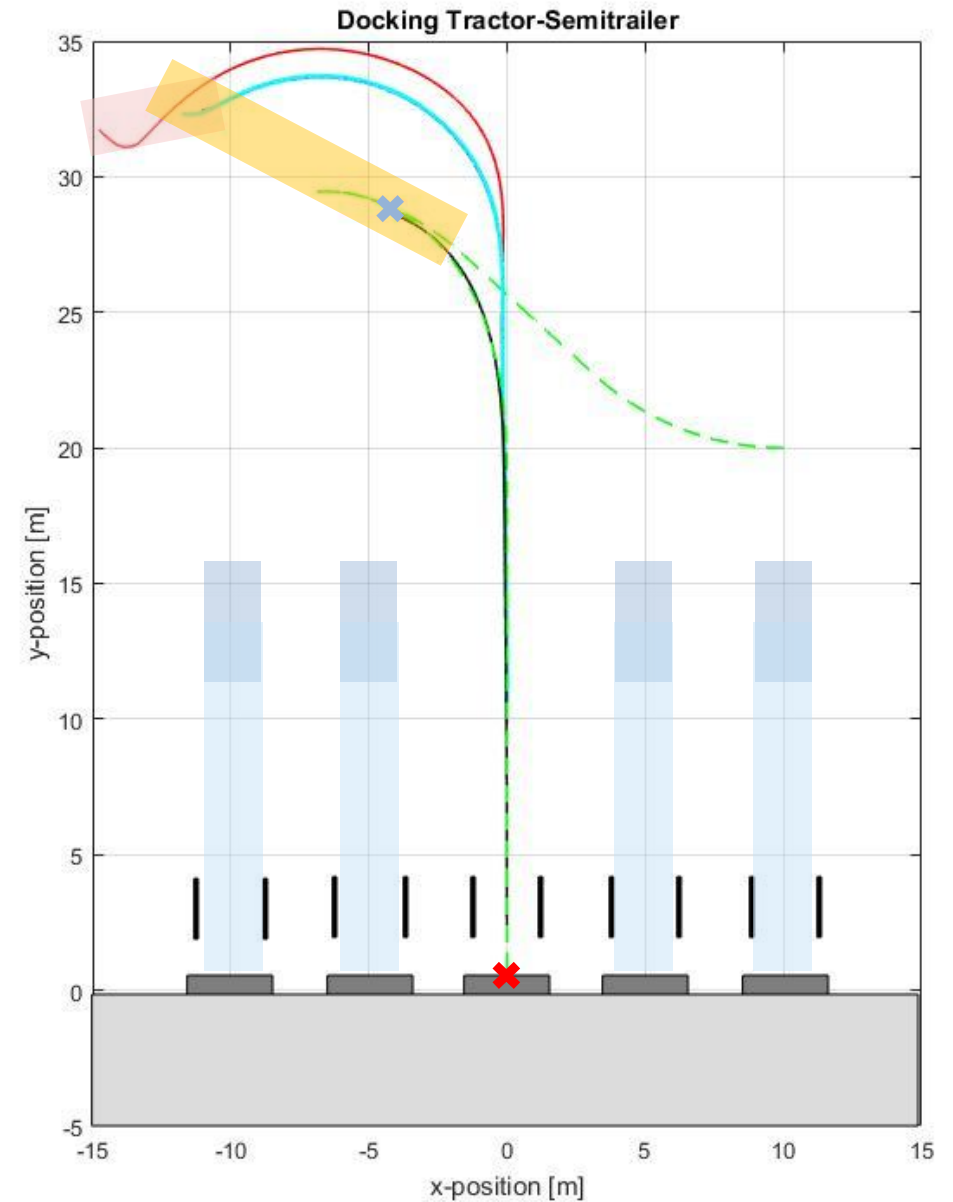
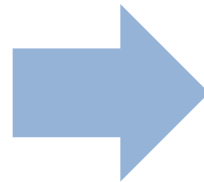
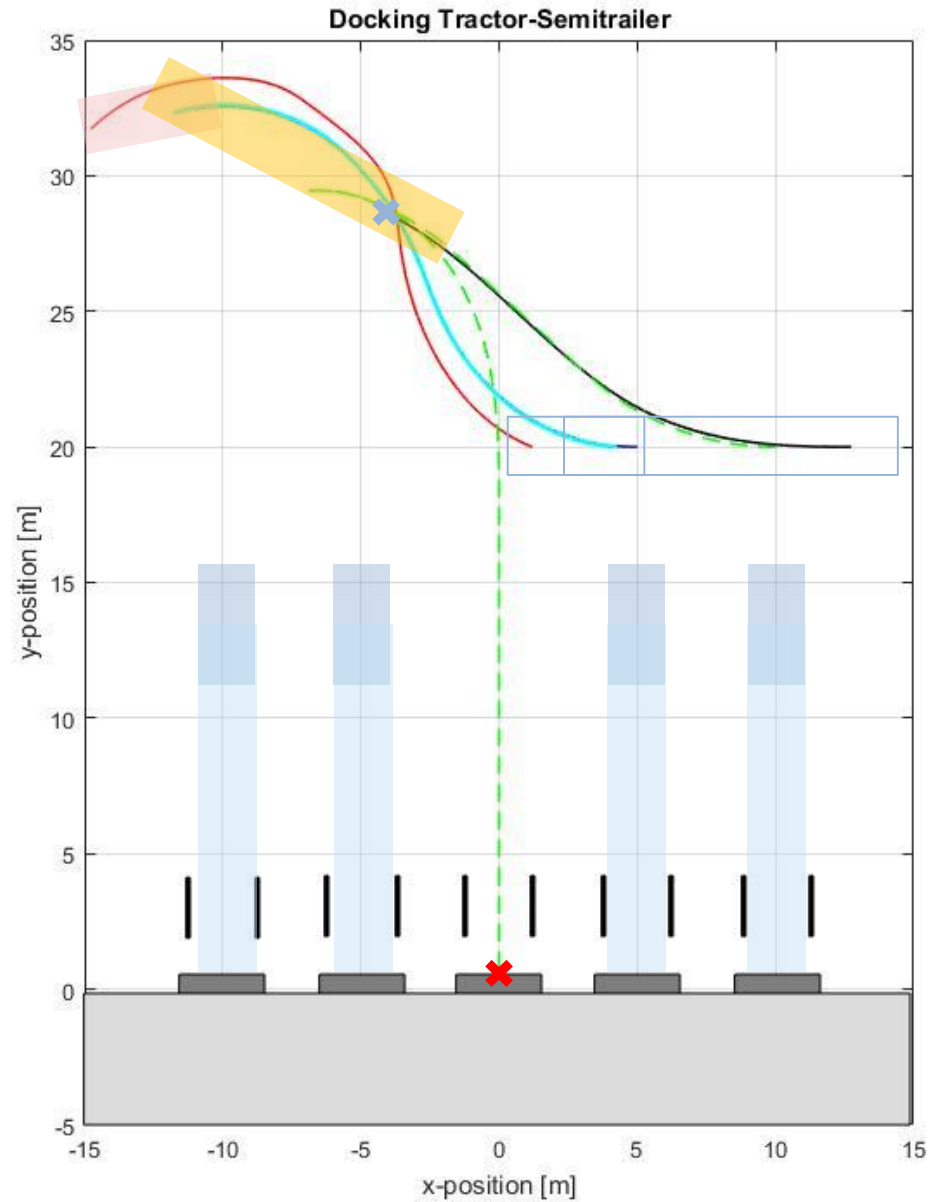
- Forward and reverse
- Minimize tracking error
- Stability



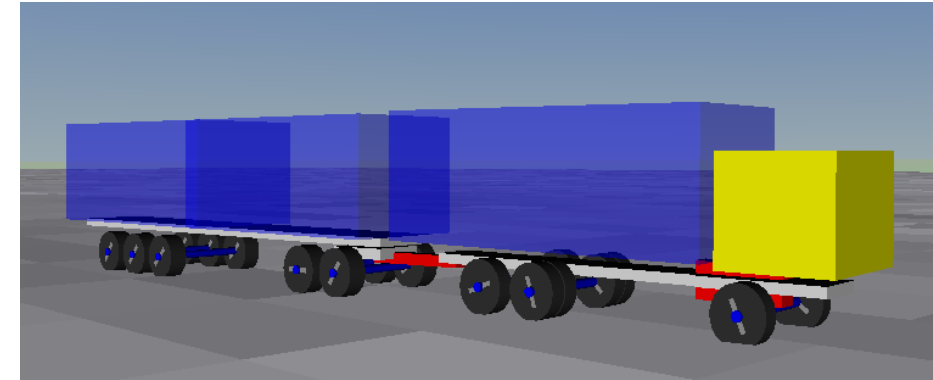
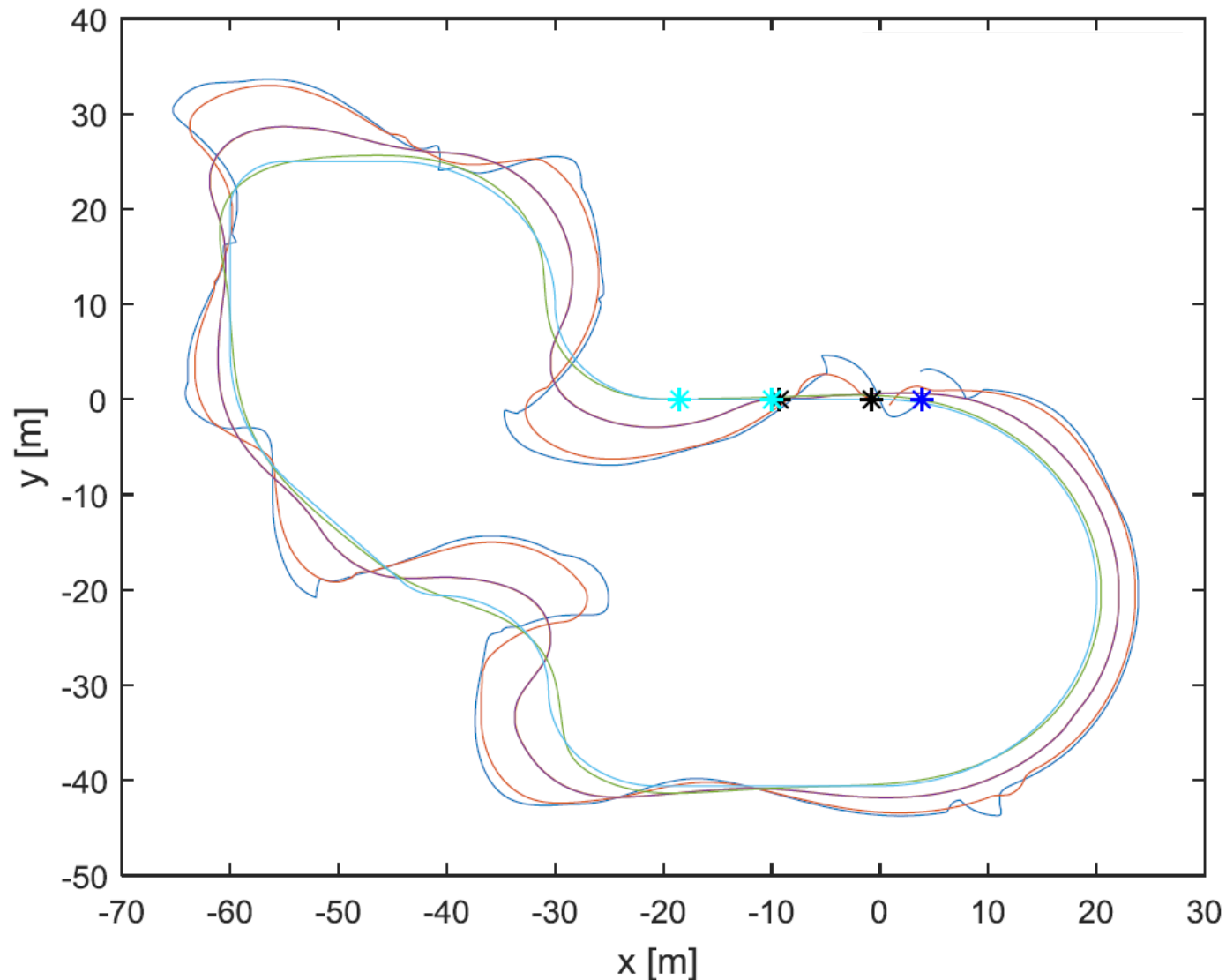








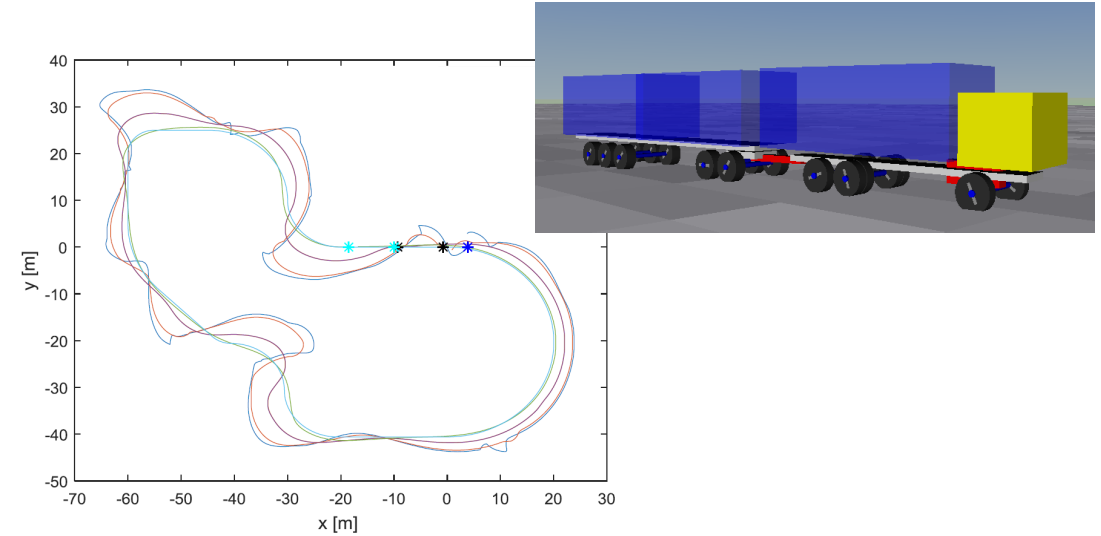
Verification on reversing of double articulated vehicle (LZV)



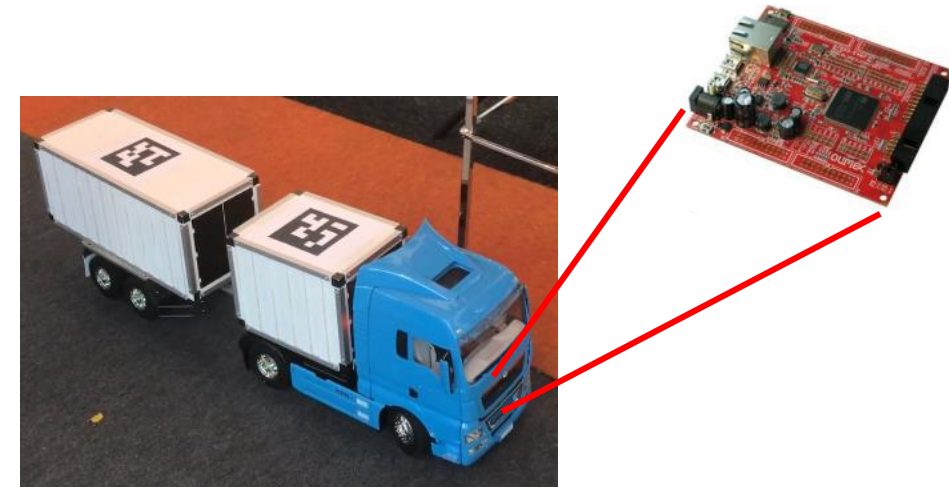
Autodocking demonstration



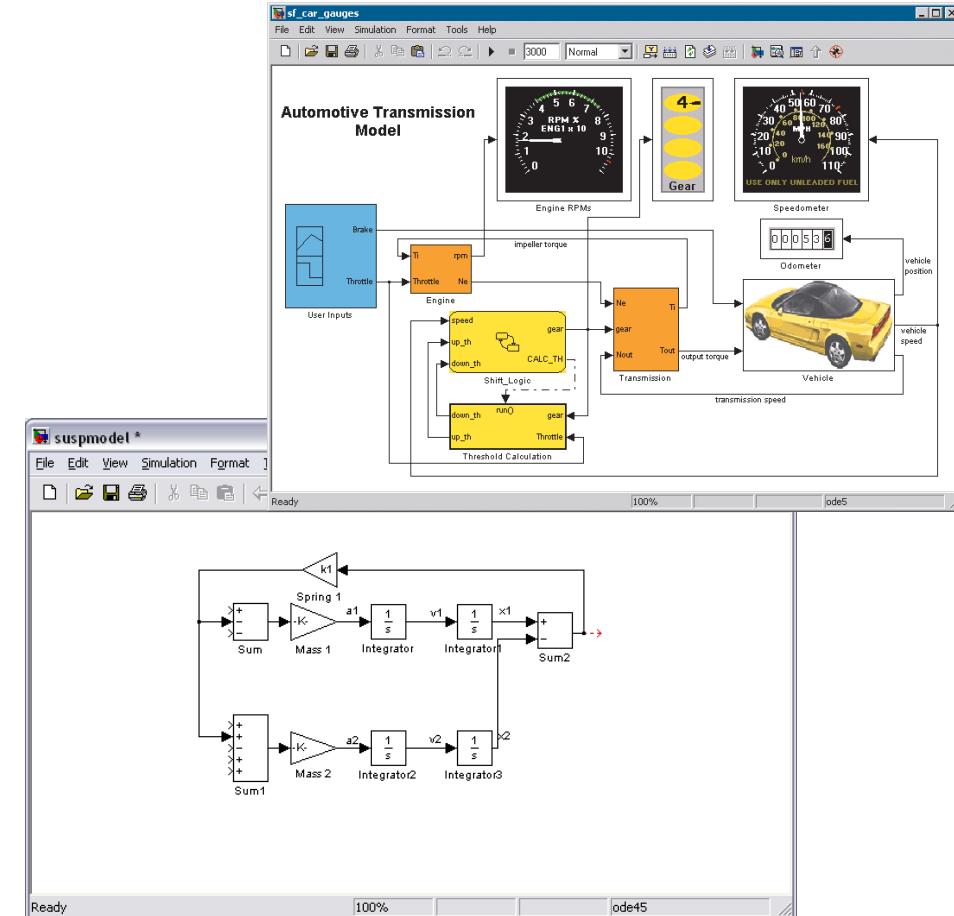
- Allows development of planner and control algorithms via simulation
 - No hardware required



- Allows control engineers to program control system
 - No programming skills required



```
1 function ye = kalmanf(A,B,C,Q,R,u,t,yv) %#eml
2 P = B*Q*B'; % Initial error covariance
3 x = zeros(size(B)); % State initial condition
4 ye = zeros(length(t),1);
5 errcov = zeros(length(t),1);
6 for i=1:length(t)
7     % Measurement update
8     Mn = P*C' / (C*P*C'+R);
9     x = x + Mn*(yv(i)-C*x); % x[n|n]
10    P = (eye(size(A))-Mn*C)*P; % P[n|n]
11    % Compute output
12    ye(i) = C*x;
13    errcov(i) = C*P*C';
14    % Time update
15    x = A*x + B*u(i); % x[n+1|n]
16    P = A*P*A' + B*Q*B'; % P[n+1|n]
17 end
```

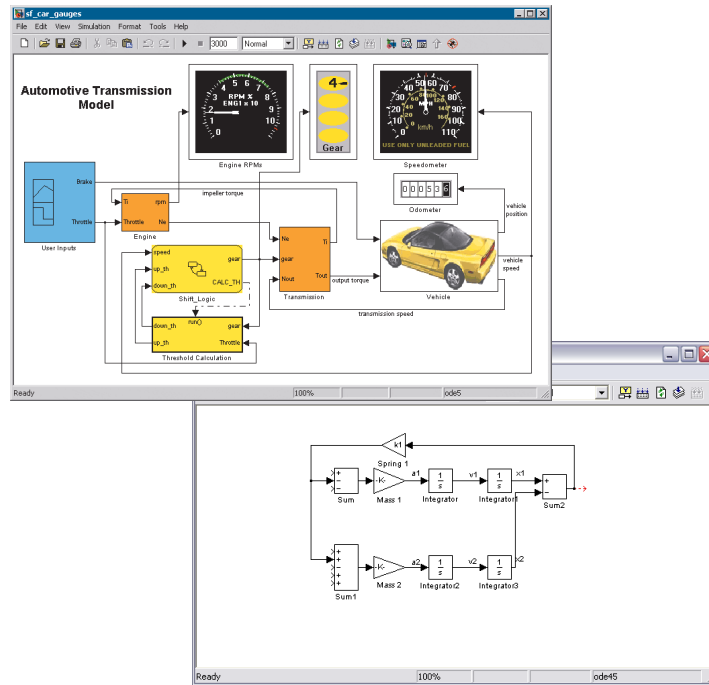


Source: Mathworks

No complex code

but intuitive blocks

Model



Specification

clear communication with customer
and between engineers (in various disciplines)

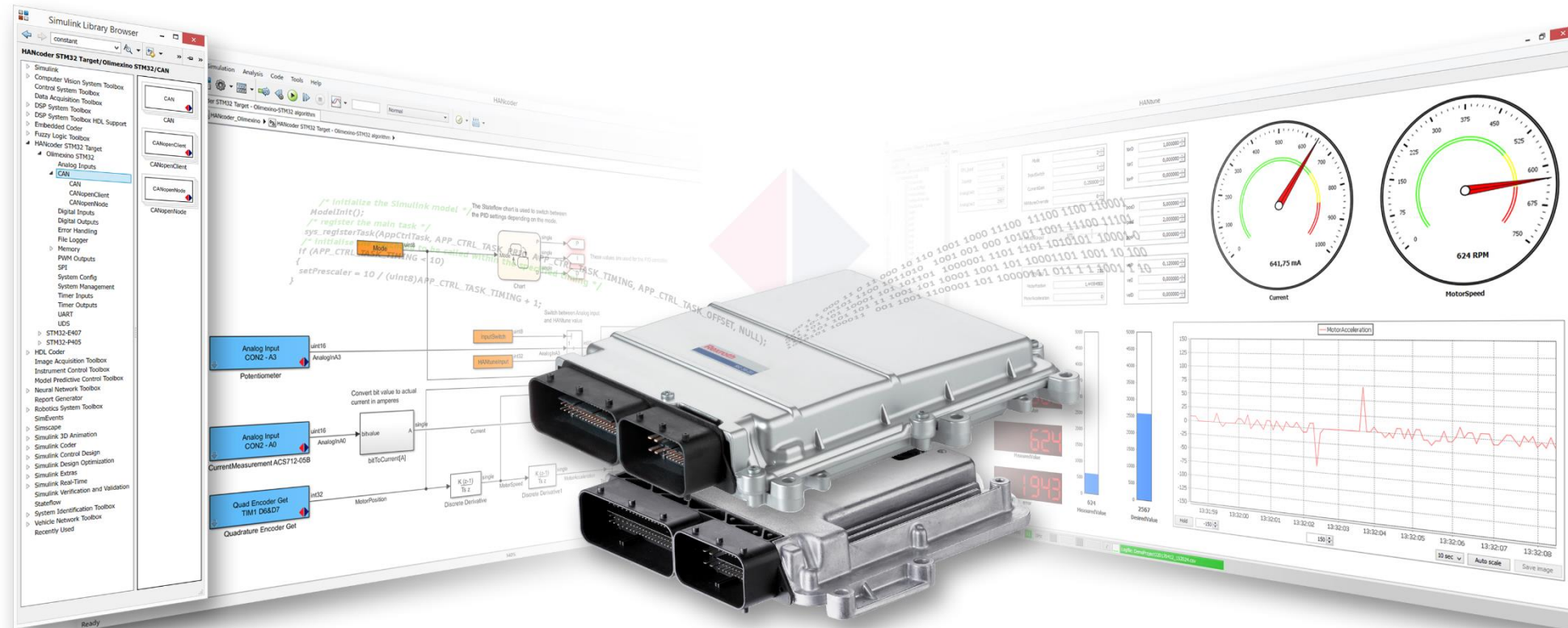
Documentation

visual , self-documenting

Implementation

can be translated in an executable program

*Basis for model based development
methodology using HAN tools*



*Various target
hardware platforms*

HAN  coder

HAN  tune

STM32 Target



RC30 Target



Prodrive GCU Target



Coming...
Linux Target



Supported hardware

Olimexino-STM32
STM32-E407
STM32-P405

Supported hardware

RC36-20/30
RC28-14/30
RC12-10/30

Supported hardware

Prodrive GCU2420

Hardware

Lely GCB
(NXP i.MX6)

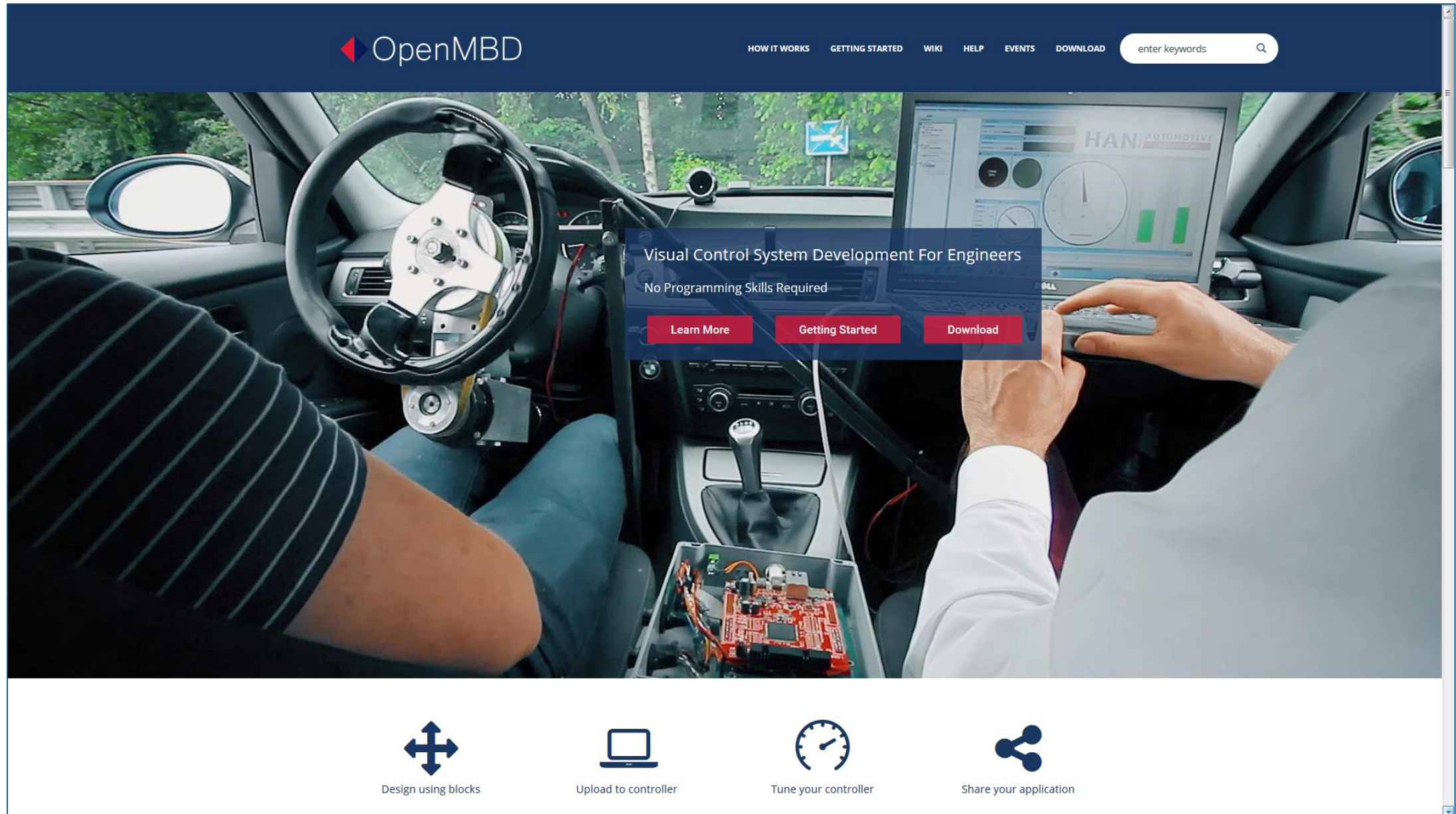
From playground to robust and full scale

STM32 Target



RC30 Target





The screenshot shows the OpenMBD website interface. The top navigation bar is dark blue with the OpenMBD logo on the left and links for HOW IT WORKS, GETTING STARTED, WIKI, HELP, EVENTS, and DOWNLOAD on the right. A search bar with the placeholder text 'enter keywords' is also present. The main content area features a large background image of a car's interior with a custom steering wheel and a laptop displaying a dashboard. A semi-transparent dark blue box is overlaid on the image, containing the text 'Visual Control System Development For Engineers' and 'No Programming Skills Required'. Below this text are three red buttons: 'Learn More', 'Getting Started', and 'Download'. At the bottom of the website, there are four icons with corresponding text: a four-way arrow icon for 'Design using blocks', a laptop icon for 'Upload to controller', a speedometer icon for 'Tune your controller', and a share icon for 'Share your application'.

OpenMBD

HOW IT WORKS GETTING STARTED WIKI HELP EVENTS DOWNLOAD

enter keywords

Visual Control System Development For Engineers

No Programming Skills Required

Learn More Getting Started Download

Design using blocks Upload to controller Tune your controller Share your application

Thank you for your attention

