ME 315 Supervised Learning Project

REACHABILITY BASED SAFE OVERTAKING

Final Presentation

PRESENTED BY
Prachit Gupta
210100111

Problem Statement

- Overtaking is one of the most complex maneuvers a vehicle can undertake
- Safety is of paramount importance here
- Combining these requires Action anticipation, intent prediction, and proactive behavior
- For autonomous vehicles AVS, we face several key challenges:

1) Accounting for uncertainty in human driver actions
2) Ensuring safety without unduly impacting planner performance
3) How to decide when to overtake and is there a way we can retreat?
4) How to do all that while ensuring human-like fluidity?

Recent Advances Enable us to deal with each of these

It's the Integration which is the most Challenging

Why Reachability

Hamilton-Jacobi (HJ) Reachability is a formal method that verifies safety in multi-agent interaction and provides a notion of safety for collision avoidance

Its advantages include :
1) Compatibility with **general nonlinear system dynamics**,
2) Formal treatment of bounded disturbances,
3) The availability of well-developed numerical tools

It gives us a set of states such that the trajectory starting in the set ends in some target set

If the target set good => reachability; the **target set bad => avoidance**

Brief Overview Reachability for collision avoidance

This is summarized by the following notions:

1)BRS / BRTS

2)differential game between 2 players

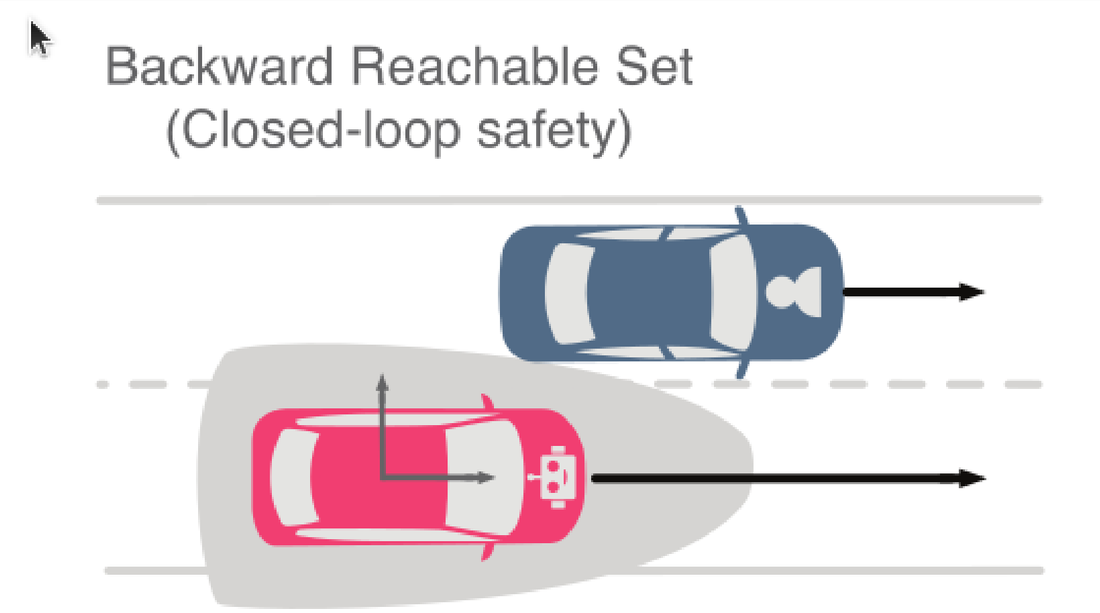
3)dynamic programming

4)level set method

5)value function

BRS/BRTS

THESE ARE SET OF ALL STATES FROM WHICH THERE EXISTS A CONTROL LAW TO REACH TARGET



Trajectories

$$\frac{d}{ds}\zeta(s;x,t,a(\cdot),b(\cdot)) = f(\zeta s;x,t,a\cdot,b\cdot)$$

$$\zeta t;x,t,a\cdot,b\cdot) = x$$

Target

$$G(t) = \{x : \exists\gamma \in \Gamma(t), \forall a(\cdot) \in A\}$$

$$\zeta(0;x,t,a(\cdot),\gamma[a](\cdot)) \in G_0$$

GAME THEORY

The problem is modeled as a differential game between player 1(robot) and player 2 (human)

for avoidance, human tries to drive the system in target and our job is to avoid going in target

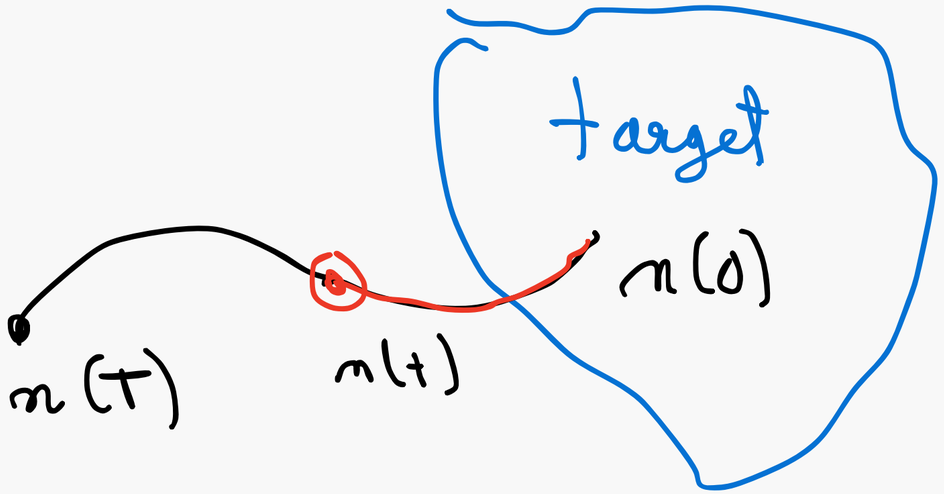
Cost **=** $J_t(x, a(\cdot), b(\cdot)) = \int_0^t c(x(s), a(s), b(s), s)\, ds + q(x(0))$

Human tries to minimize this cost while robot tends to maximize the same

Optimal Value of the cost = $G(t, x) = \inf_{\gamma \in \Gamma(t)} \sup_{a(\cdot) \in A} J_t(x, a(\cdot), \gamma[a](\cdot))$

Cost to go!!

Dynamic Programming



Intuitively

V(t+ delt) = V(t) + incremental term

THIS NOTION IS CAPTURED BY HJI PDE:

$$\frac{d}{dt}V(t,x) + H(t,x,\nabla V(t,x)) = 0, V(0,x) = q(x)$$

$$H(t,x,\lambda) = \max_{a\in A}\min_{b\in B} c(x,a,b,t) + \lambda \cdot f(x,a,b).$$

OPTIMAL CONTROL MINIMIZING H is

$$a^*(t,x) = \arg\max_{a\in A}\min_{b\in B} c(x,a,b,t) + \lambda \cdot f(x,a,b).$$

level set method

Numerically solve HJI PDE by defining a level function as zero sublevel of target set

There exists a level set function as a zero sublevel of the target set G0 i.e

$$x \in G_0 \iff g(x) \leq 0$$

So the cost function becomes

$$J_t(x, a(\cdot), b(\cdot)) = g(x0))$$

System attains target set (here collides) if

$$J_t(x, a(\cdot), b(\cdot)) \leq 0$$

BRTS/Avoid set

$$G(t) = \{x : G(t, x) \leq 0\}$$

Bang bang control

OPTIMAL CONTROL = here set of control actions which just avoids the BRS maneuver through boundary

To find optimal control we use the PMP principle which basically states that the optimal control sequence is the one that minimizes the hamiltonian

$$H(t, x, \lambda) = \max_{u \in U} \left[L(t, x, u) + \lambda^T f(t, x, u) \right]$$

where $\dot{\lambda}(t) = -\frac{\partial H}{\partial x}(t, x(t), \lambda(t))$

So this is only possible if either u makes the derivative of H go to 0 or the **minimum values occur at extreme points** which leads us to

Bang bang control = control input discontinuously jumps between umin and umax

Computation of BRTS(values)

1) Define relative dynamics

2)determine control law to compute optimal control

3)solve the partial differential equations

4)obtain value function as a solution

Dynamics

4 Dimensional KBM model for robot (more accurate modeling

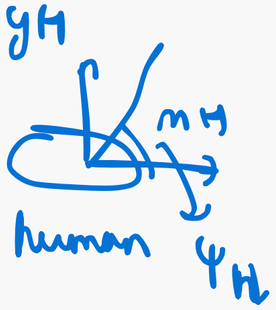
$$
\begin{aligned}
\dot{x} &= v\cos(\psi+\beta) \\
\dot{y} &= v\sin(\psi+\beta) \\
\dot{\psi} &= \frac{v}{l_r}\sin(\beta) \\
\dot{v} &= a \\
\beta &= \tan^{-1}\left(\frac{l_r}{l_f+l_r}\tan(\delta_f)\right)
\end{aligned}
$$

4d Curvature car for human (to give more freedom)

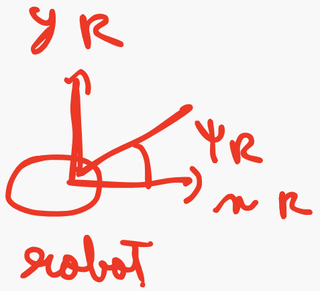
$$
\dot{x}_{\mathrm{H}} = \begin{bmatrix} v_{\mathrm{H}}\cos\psi_{\mathrm{H}} \\ v_{\mathrm{H}}\sin\psi_{\mathrm{H}} \\ \omega \\ a \end{bmatrix}
$$

Relative dynamics

to compute brs, we transform the state space **wrt to robot frame with origin attached to robot center** giving us 5D RELATIVE dynamics



$$u_{\mathrm{H}} = (a_{\mathrm{H}}, \omega_{\mathrm{H}})$$



$$u_{\mathrm{R}} = (a_{\mathrm{R}}, \delta_{\mathrm{f}})$$

$$x_{\mathrm{rel}} = (px_{\mathrm{rel}}, py_{\mathrm{rel}}, \psi_{\mathrm{rel}}, v_{\mathrm{R}}, v_{\mathrm{H}})$$

$$\dot{x}_{\mathrm{rel}} = vR\frac{py_{\mathrm{rel}}}{l_{\mathrm{r\ rel}}}\sin(\beta_{\mathrm{r}}) + v_{\mathrm{H}}\cos(\psi_{\mathrm{rel}}) - v_{\mathrm{R}}\cos(\beta_{\mathrm{r}})$$

$$\dot{y}_{\mathrm{rel}} = -vR\frac{pyx_{\ \mathrm{rel}}}{l_{\mathrm{r\ rel}}}\sin(\beta_{\mathrm{r}}) + v_{\mathrm{H}}\sin(\psi_{\mathrm{rel}}) - v_{\mathrm{R}}\sin(\beta_{\mathrm{r}})$$

$$\dot{\psi}_{\mathrm{rel}} = \omega_{\mathrm{H}} - \frac{v_R}{l\mathrm{r}\sin(\beta_{\mathrm{r}})}$$

$$\dot{v}_R = a_R$$

$$\dot{v}_H = a_H$$

Other parameters

Target set for now is set of states where robot and human are within a rectangle of 6 x 4 metres i.e xrel and yrel constrained in a rectangle about origin

Controls

Are system dynamics are **non linear and non affine wrt control**

Fortunately, every non linearilty and non affinity
eg cos(phi), atan(control) or tan(atan(control)
are all monotonically increasing in given domain so notion of extreme control is the optimal control is still valid

Validated experimentally

Toolboxes

Helper Oc = 1)give more freedom in terms of defining own dynamics and control law

2)Easier to implement in Matlab but 100 times slower

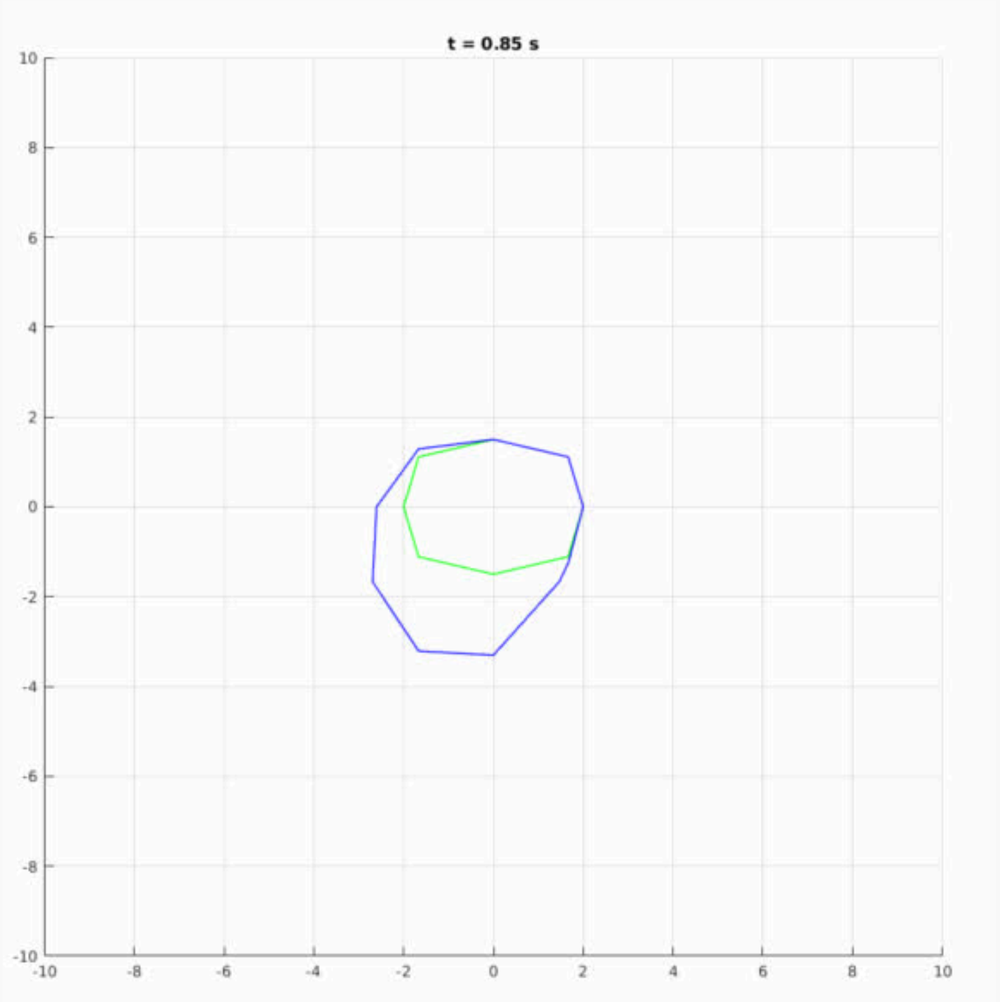
Hj Reachability = 1)best in terms of ease of implementation

2)100 x faster

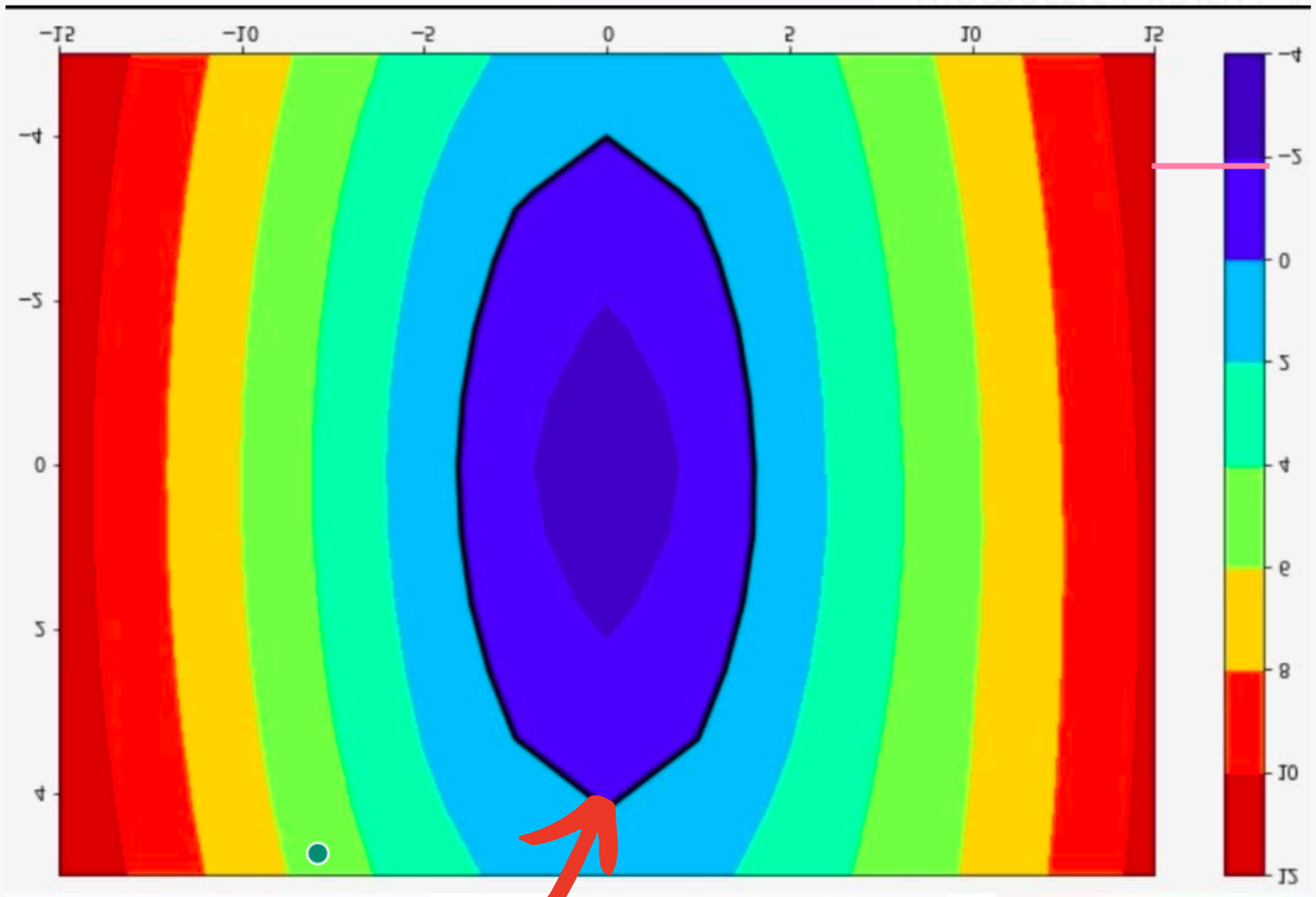
Unfortunately only works for affine dynamics right now, lax Friedrichs approximations leads to more accurate BRTS shape but requires user defined notions of lower and upper bounds (difficult for our dynamics)

BEACLS = C implementation of helper OC , best choice only if I was proficient in C , fastest

Results



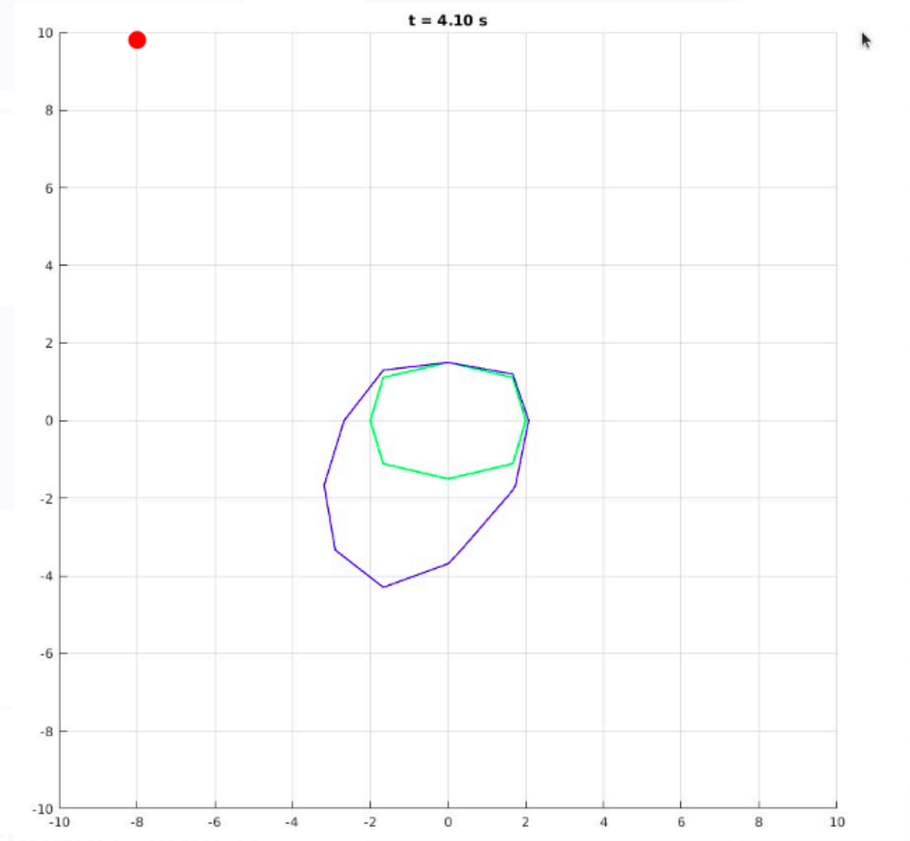
Computation helper oc

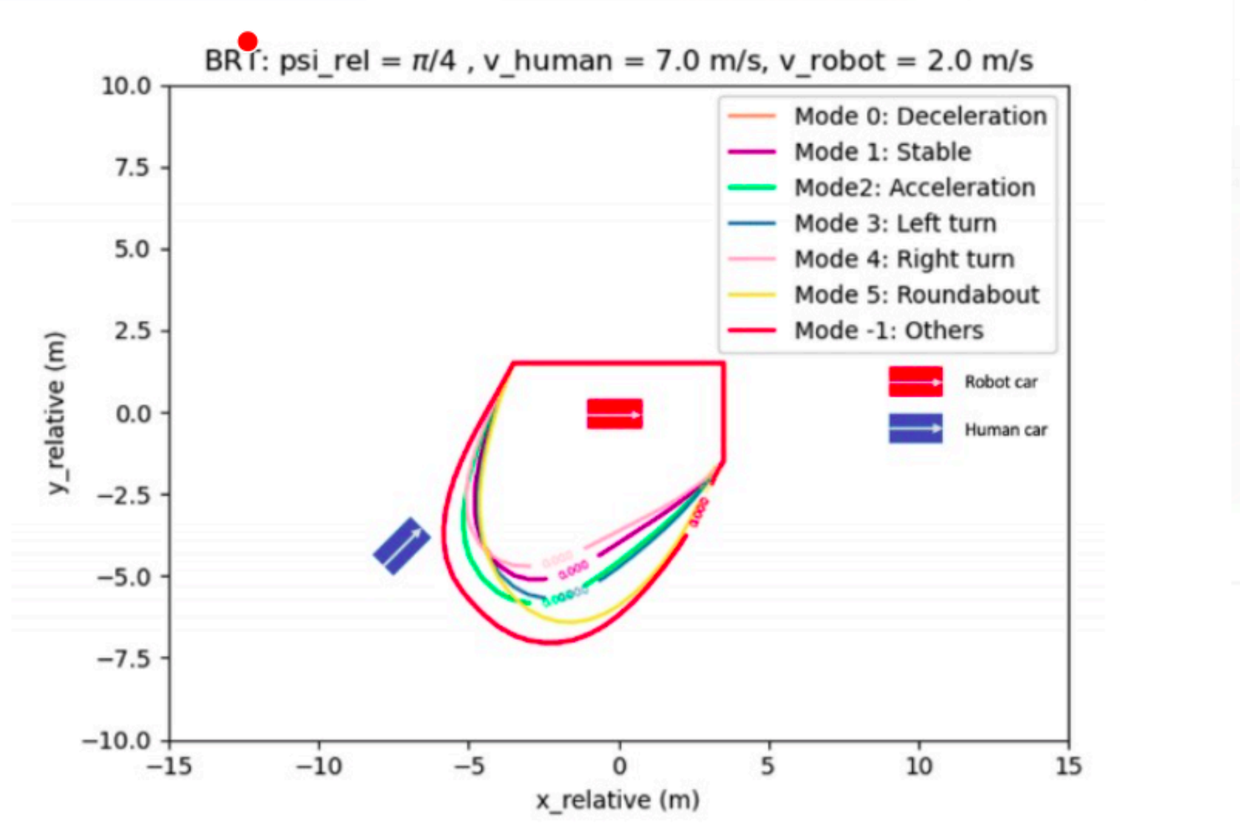


Heat Map

2D SLICE OF BRT

Verification





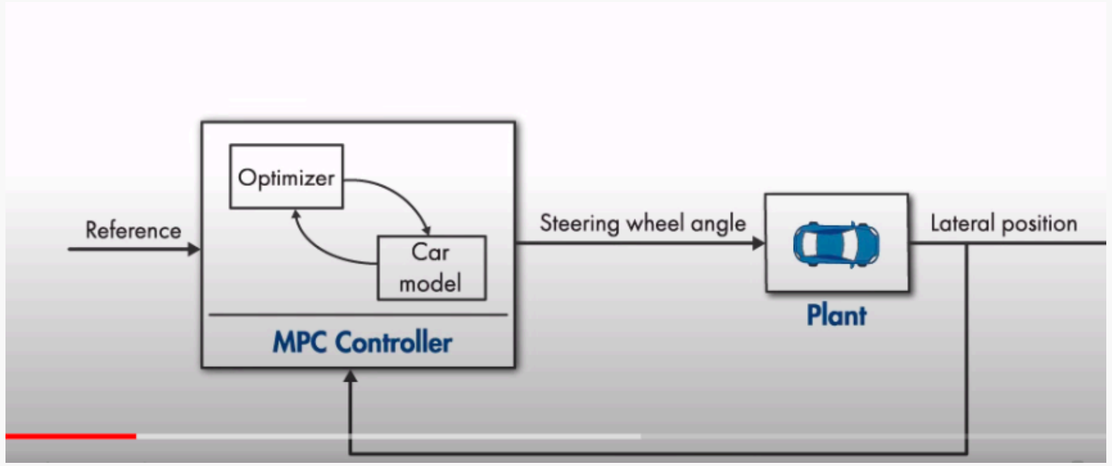
SO WE HAVE **VALUE(SCALAR) AT EVERY POINT** ON THE GIVEN HORIZON

BRT or **avoid set is where this value assumes negative** values

MPC

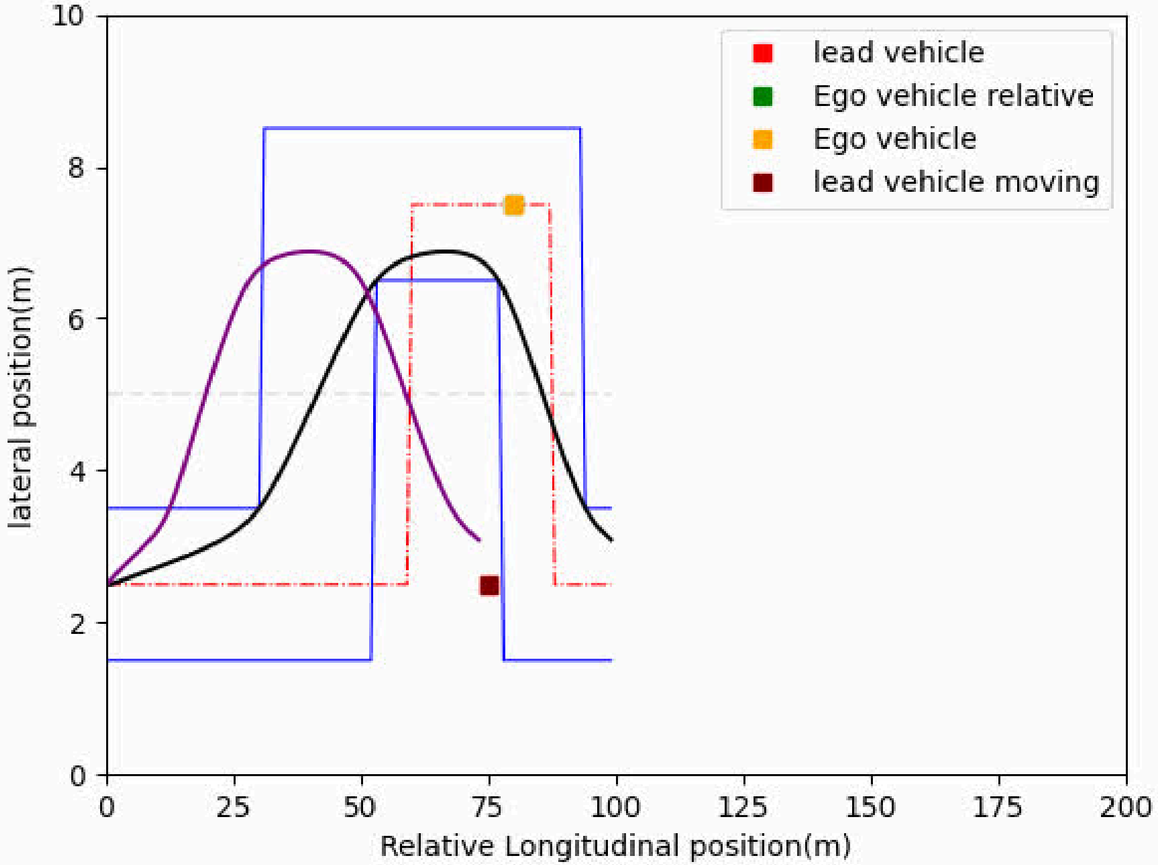
MPC works by **Continuously predicting future system behavior**, evaluating different control actions, and selecting the optimal one that minimizes a predefined cost function,

Prediction is made for entire horizon but implementation only for a time step



$$
\begin{aligned}
\min_{\mathbf{U}} \quad & \sum_{i=0}^{H_p} (\mathbf{z}_i - \mathbf{z}_{ref,i})^T Q (\mathbf{z}_i - \mathbf{z}_{ref,i}) + && \text{(4a)} \\
& \sum_{i=0}^{H_p-1} \left[(\mathbf{u}_i - \mathbf{u}_{i-1})^T \bar{R} (\mathbf{u}_i - \mathbf{u}_{i-1}) + \mathbf{u}_i^T R \mathbf{u}_i \right] \\
\text{s.t.} \quad & \mathbf{z}_0 = \mathbf{z}(t), \mathbf{u}_{-1} = \mathbf{u}(t - t_s) && \text{(4b)} \\
& \mathbf{z}_{i+1} = f(\mathbf{z}_i, \mathbf{u}_i), \ \ i = 0, \ldots, H_p - 1 && \text{(4c)} \\
& \mathbf{u}_{min,i} \leq \mathbf{u}_i \leq \mathbf{u}_{max,i}, \ \ \forall i && \text{(4d)} \\
& \dot{\mathbf{u}}_{min,i} \leq \frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\mathrm{t}_d} \leq \dot{\mathbf{u}}_{max,i}, \ \ \forall i \backslash 0 && \text{(4e)} \\
& \dot{\mathbf{u}}_{min,i} \leq \frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\mathrm{t}_s} \leq \dot{\mathbf{u}}_{max,i}, \ \ i = 0 && \text{(4f)}
\end{aligned}
$$

MPC for overtaking

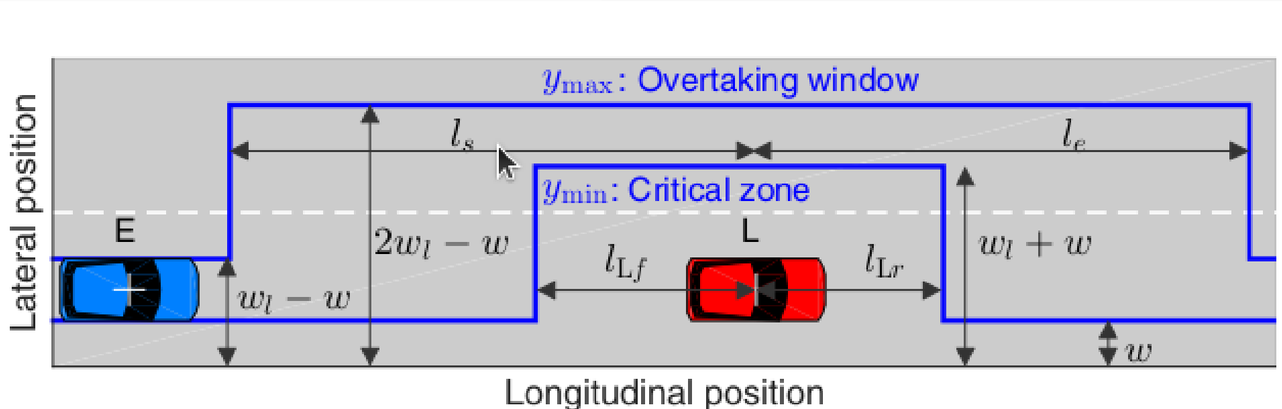


Explicit corridor constraints

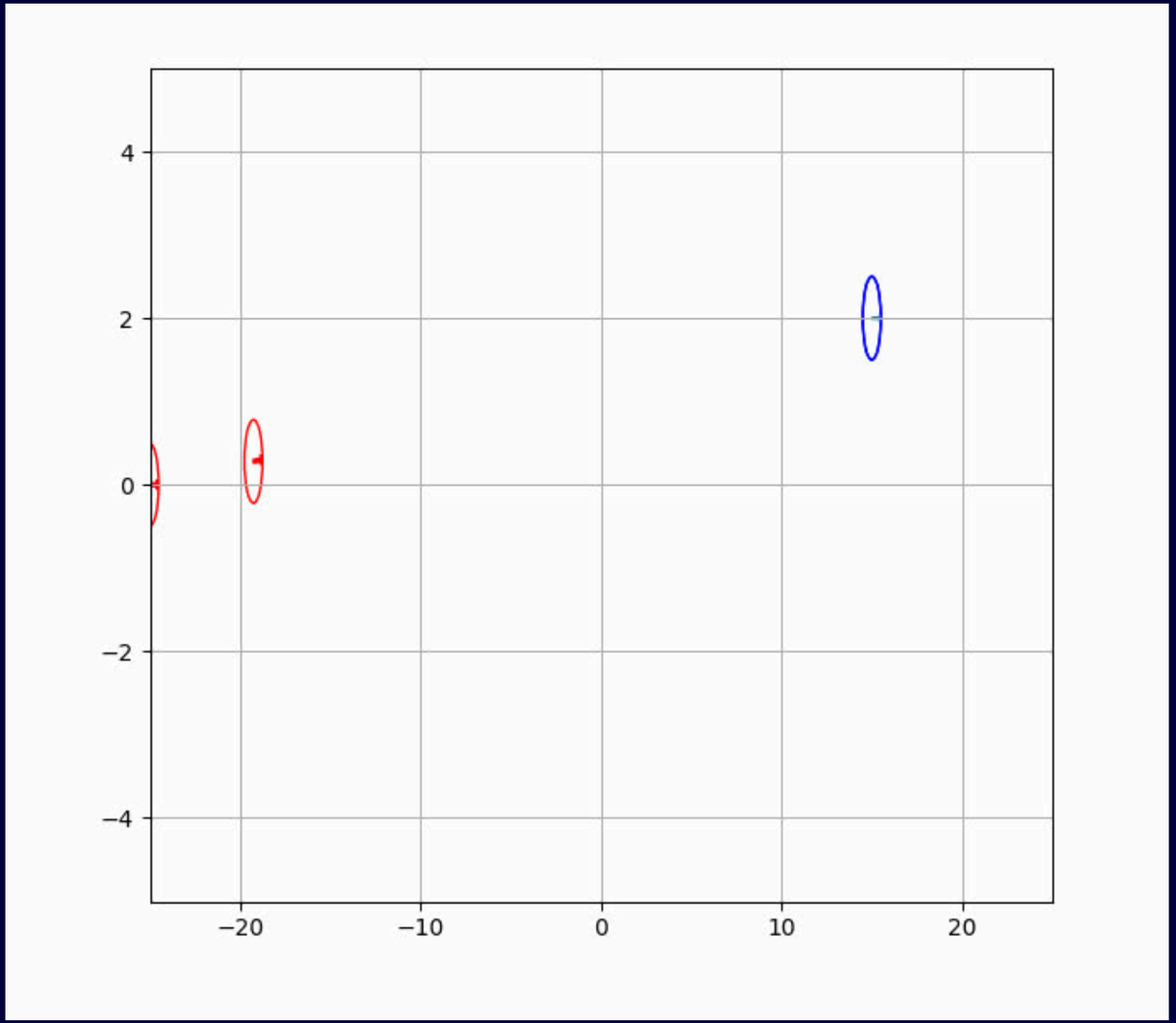
Not robust to human actions

Forcing overtaking by giving a reference traajectory

Impractical reference trajectory



MPC FOR OUR ROBOT DYNAMICS3



GOAL IS TO REACH THE BLUE HOOP

SMOOTH TRAJECTORIES FORMED MINIMIZING DISTANCE OF START(Recursively updated) and END

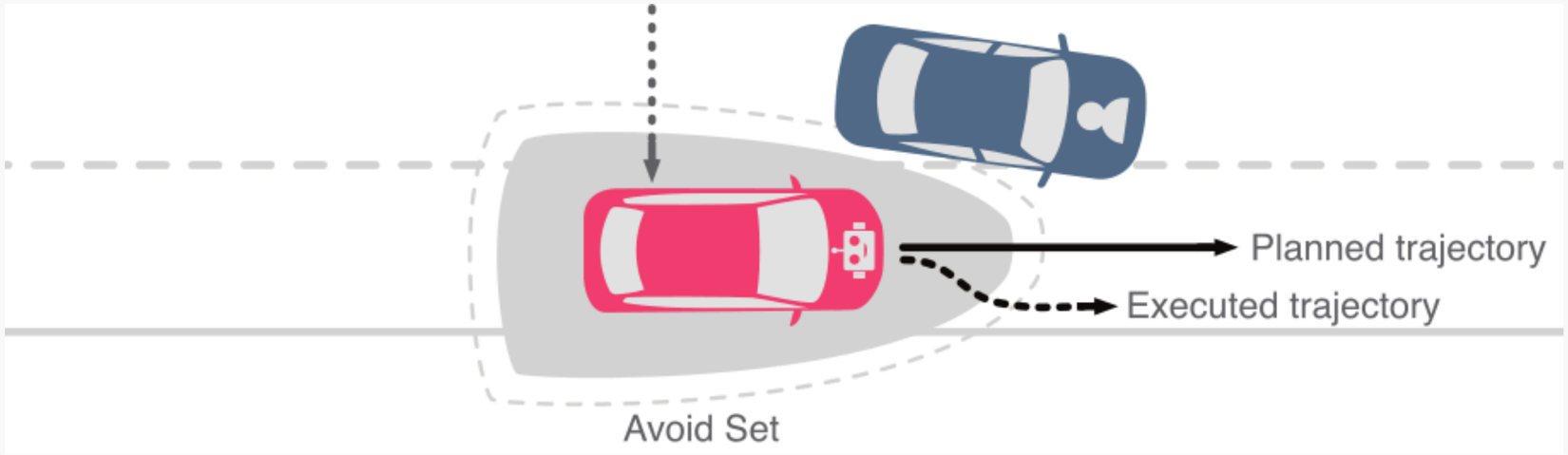
Integration

We have:

1) a robust MPC framework in Casadi to reach a desired state

2) Pre-computed value functions giving us the value at a given relative state

3)simulation of dummy human motion (for now moving at a constant speed)



Approach

Accessing values in MPC framework :

1)Generated a look-up table by fitting a 5 dimensional spline polynomial between relative state and corresponding values

Casadi supports look-up tables so within optimization framework:

2)calculated relative dynamics using human and robot state variables (robot is a decision variable and human merely a parameter)

3)Iteratively calling look-up table to implicitly obtain value function at relative state

Formulation

$$\min_{U,X} \sum_{k=0}^{N} \left[(X_k - X_f)^T R (X_k - X_f) + U_k^T Q U_k + \mathbf{value}(X_{rel}) \right.$$

subject to

$$u = [u_{min}, u_{max}]$$

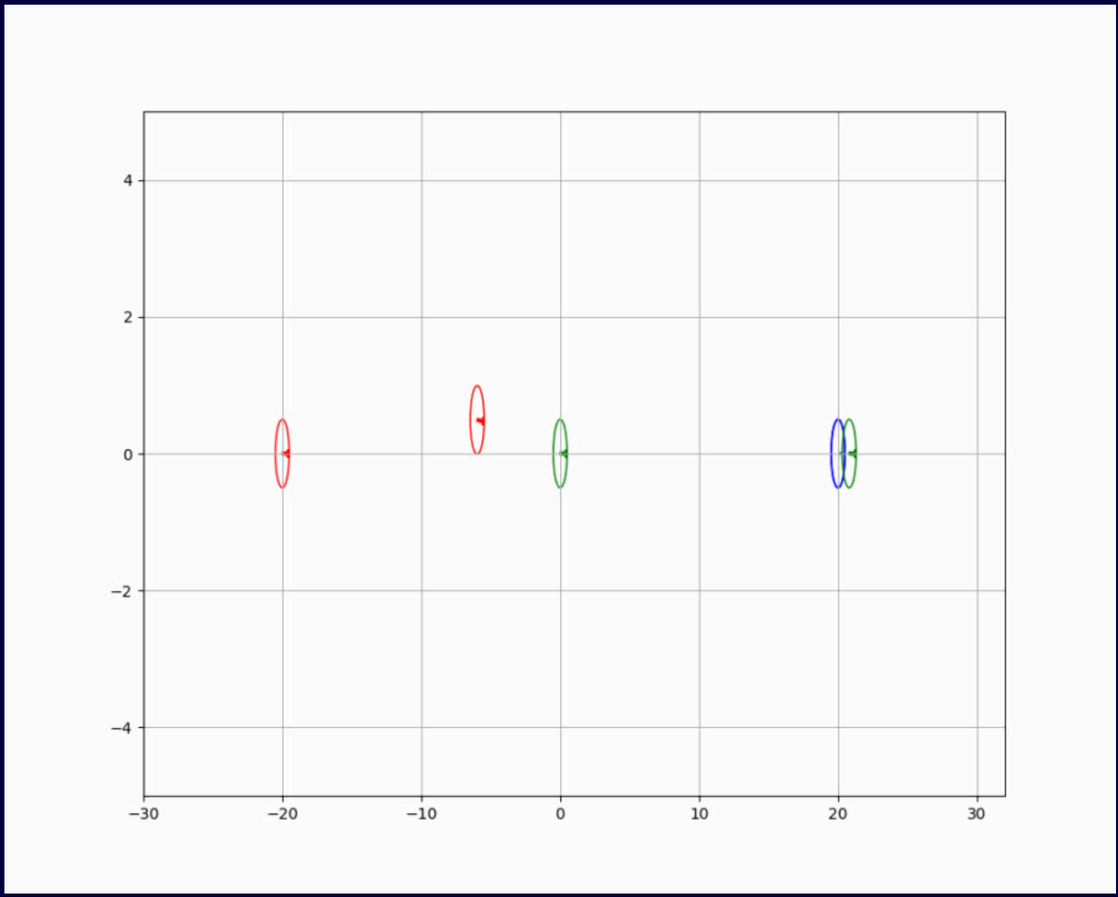
$$X = [X_{min}, X_{max}]$$

$$X_0 = X_{initial}$$

$$\dot{x}_{\text{robot}} = f(x_{\text{robot}}, u_{\text{R}}, u_{\text{H}})$$

and

$$\mathbf{value}(X_{rel}) \geq \epsilon$$

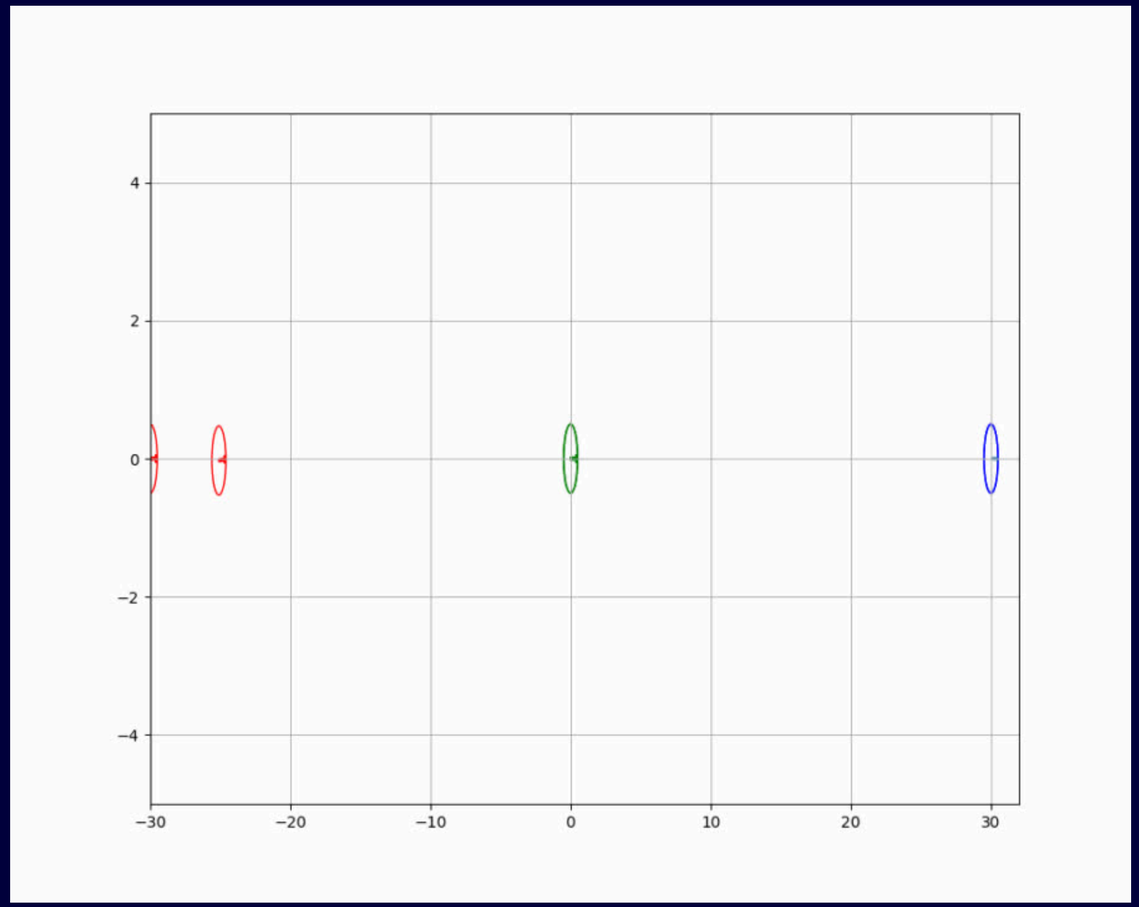


Initial conditions of human don’t guarantee safe overtaking

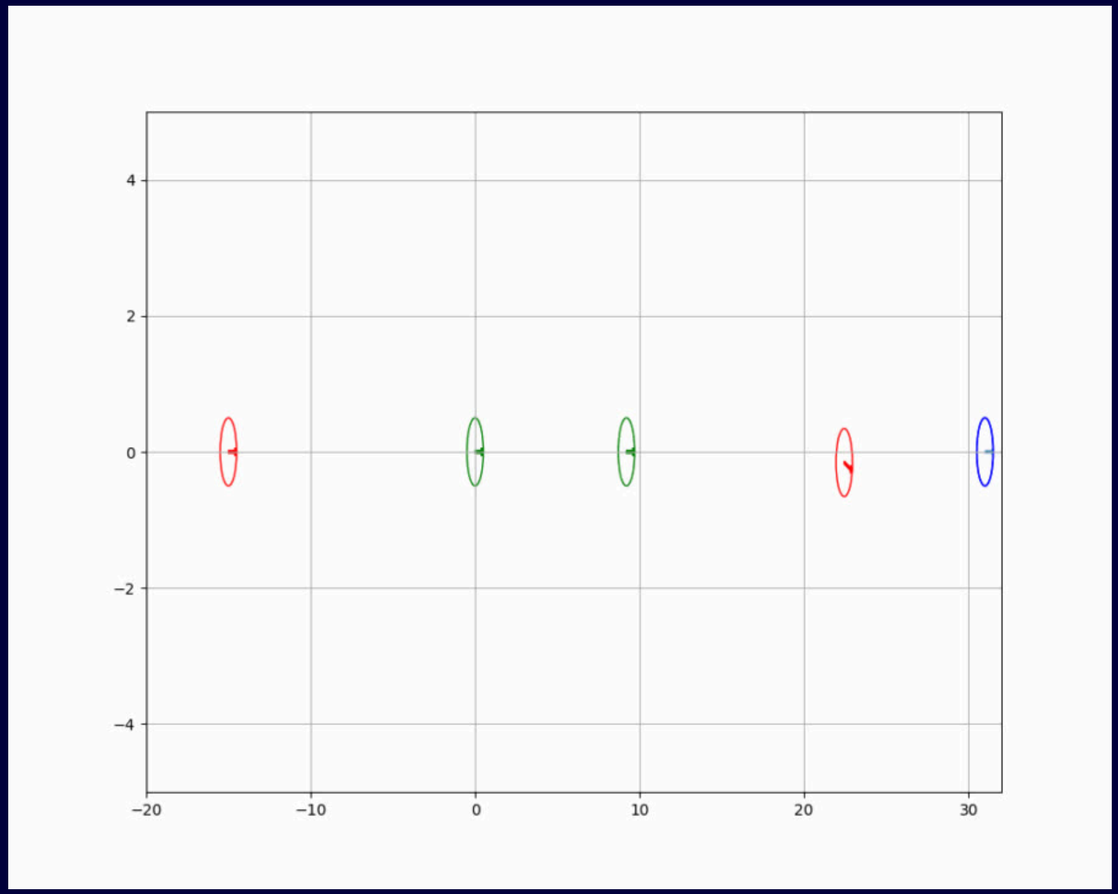
so the robot waits till value functions become favourable to reach desired goal

Results

Demonstrating effect of value function



when initial conditions favorable, static human robot safely overtakes the lead vehicle



when initial conditions favorable, slow-moving human robot safely overtakes the lead vehicle

Implementation gaps

1) State not converging to desired final position due to the value function associated with that state : requires defining overtaking horizon (for which value functions have been computed) as a subset of prediction horizon

2) weak notion of the value function in objective, need an analytical function that handles tradeoff between safety and minimal deviation

3) random jerky motion , need to punish jerk in objective as well

4) better way to enforce overtaking rather than fixing a desired state

Novel Contributions

Perform overtaking **only when** its safe and feasible

No arbitrary reference trajectory needed , automatically generate safe trajectory

Didn’t find any solely reachability based trajectory generator in literature

Minimally intervening to smooth motion unlike existing switching controller based approaches where the entire control stack is switched to emergency bang-bang control in case of safety breach

SOFTWARE CONTRIBUTIONS

Developed a **VALUE-FUNCTION-GENERATOR** package based on helperOC with documentation on :

1)Generating value function with custom relative 5d state space

2)Generating a CASADI based lookup-table from the value functions

3)Scripts for integrating value functions in any MPC-based control stack

4)Scripts for animation and visualization

ValueFunctionGenerator

Research gaps and proposed solutions

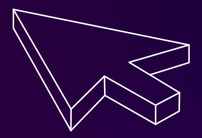
Conservative BRTS , considers all possible motions of humans which lead to overly conservative trajectories (jerky deviations ensuring safety even when it's completely safe in the real world

Solutions Proposed in literature = 1) Some **game theoretic human model on human** to generate BRTS only corresponding to predicted motions not all possible motions
2) **Multiple BRTS** each considering a separate action by Human and then **dynamically switching between BRTS** online

The weak notion of the target set solely based on distance, the collision might be inevitable even when vehicles are significantly apart

Open research problem as far as I know

Bibliography



1	HJI theoritical background - Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances Somil Bansal*, Mo Chen*, Sylvia Herbert* and Claire J. Tomlin
2	Kinematic bicycle model based MPC- Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design Jason Kong 1 , Mark Pfeiffer 2 , Georg Schildbach 1 , Francesco Borrelli 1
3	Predictive Cruise Control with constraints on lateral dynamics **Predictive cruise control with autonomous overtaking** **Nikolce Murgovski and Jonas Sjöberg**
4	Helper Oc tool box - https://github.com/HJReachability/helperOC
5	Infusing reachability in controllers - On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions

Thank You