

ONLINE CONSTRAINED MODEL-BASED REINFORCEMENT LEARNING

Benjamin Mann Niekirk¹, Andreas Damianou², Benjamin Rosman^{1,3}

1. University of the Witwatersrand, 2. Amazon.com (work done while at the University of Sheffield), 3. Council for Scientific and Industrial Research



MOTIVATION

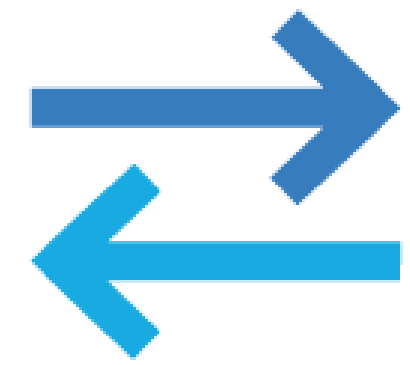
Reinforcement learning has made impressive strides in recent years, but its application to robotic systems still poses a number of challenges:

Consider an autonomous racing task...


Real-time planning.
Decisions must be made within a limited time budget – often in the range of a few milliseconds



Data efficiency.
Collecting experience, or evaluating policies, with a robot in the loop can be costly and time consuming. Poorly modelled dynamics, due to a slow learning rate, may result in unstable trajectories or collisions.



Continuous state and control spaces.
The dynamics of robotic systems are naturally modelled in terms of continuous variables – position, velocity, orientation, steering angle, throttle, etc.



Safety.
For safe operation, the system must make decisions under physical constraints – from actuator limitations to the boundaries defined by the track.

TO TACKLE THESE CHALLENGES WE COMBINE:

- A planner based on **RECEDING HORIZON CONTROL**.
- A dynamics model learned using **SPARSE SPECTRUM GAUSSIAN PROCESSES**.

RECEDING HORIZON CONTROL

Trajectory optimization.
The idea of receding horizon control (RHC) is to plan over a finite horizon by iteratively solving an optimization problem. Given a measurement of the current state, a trajectory through the state-control space is calculated, and the first step of the control signal is applied to the system. The horizon is then shifted forward and the process repeats.

$$\min_{\mathbf{x}, \mathbf{u}} \quad h(\mathbf{x}_N) + \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}_k, \mathbf{u}_k), \quad \text{--- Cost function.}$$

subject to

$\mathbf{x}_0 = \hat{\mathbf{x}}_0,$

$\mathbf{x}_{k+1} = \mathbf{F}_k(\mathbf{x}_k, \mathbf{u}_k),$

$\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \bar{\mathbf{u}},$

$\underline{\mathbf{x}} \leq \mathbf{x}_k \leq \bar{\mathbf{x}},$

$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0}.$

Initial condition.

Dynamics model.

Box constraints.
Used to specify joint or actuator limitations.

Non-linear constraints.
Used to specify boundaries or obstacles..

Sequential quadratic programming.

To solve the optimization problem efficiently, the dynamics model, cost function and constraints are linearized about a nominal trajectory. The trajectory can then be improved by solving a sequence of quadratic programs.

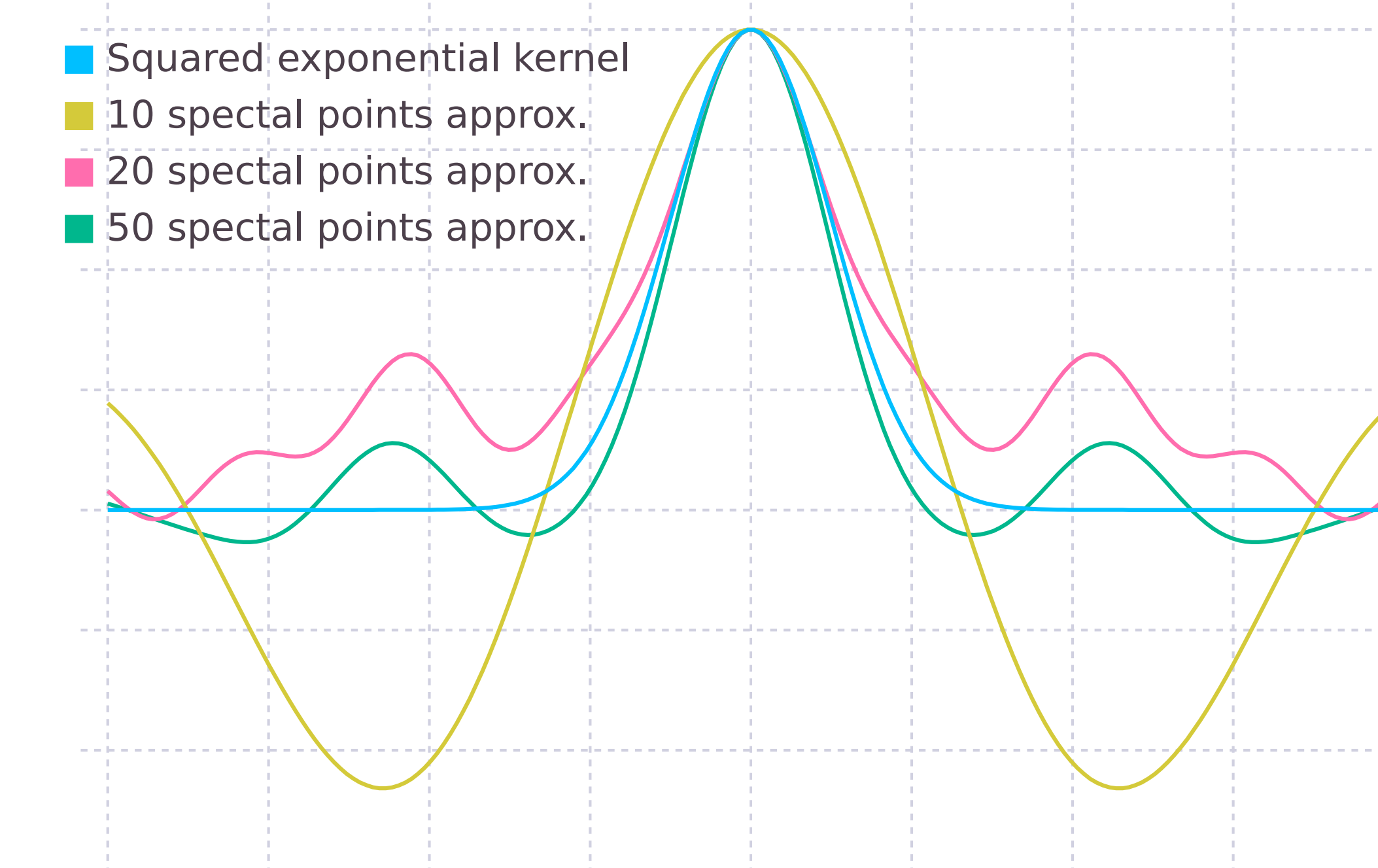
We use FORCES¹ as the stage-wise solver for the quadratic programs. FORCES is an interior-point method tailored for problems arising in receding horizon control.

Why RHC?

- Instead of explicitly maintaining a representation of a policy or value function, the control is recalculated (over the shifting horizon) at each time step. Replanning may help reduce inaccuracies introduced by model error.
- Hard constraints can be explicitly specified, allowing agents to safely learn by avoiding dangerous regions of the state and control spaces.

SPARSE SPECTRUM GAUSSIAN PROCESSES

Learning the dynamics.
Given a training set of observed state transitions a model of the underlying dynamics is learned using sparse spectrum Gaussian processes².

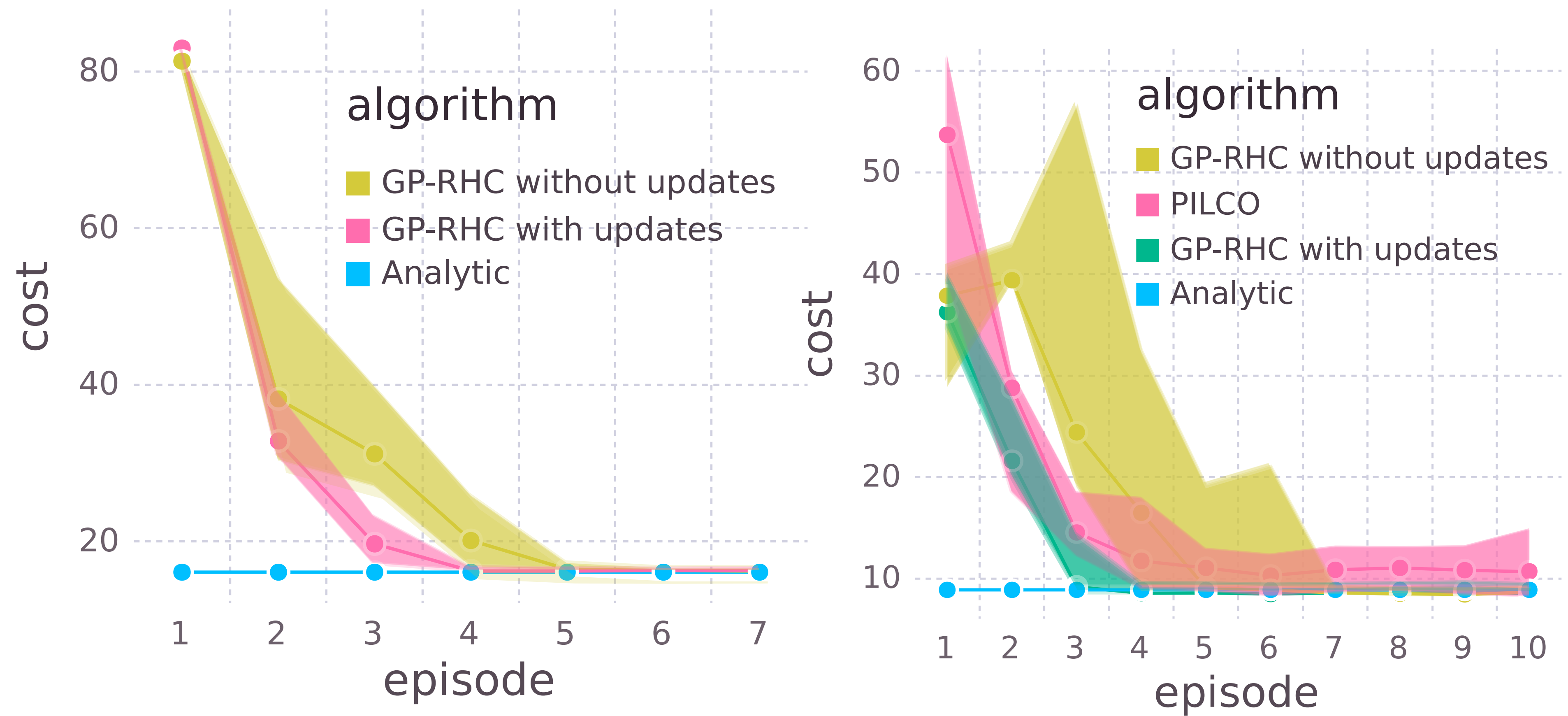


Sparse spectrum approx.
The kernel function of a stationary Gaussian processes can be represented in terms of its power spectrum. By taking a finite number of samples from the power spectrum, the kernel can be approximated efficiently.

Why sparse spectrum GPs?

- Reduced computational and memory requirements compared to full Gaussian process regression. This allows the model to scale up to larger data sets.
- Online data can be incorporated through incremental updates resulting in more data efficient and robust learning.
- The learned model can be efficiently linearized to form the sequential quadratic programs allowing for real-time applications.

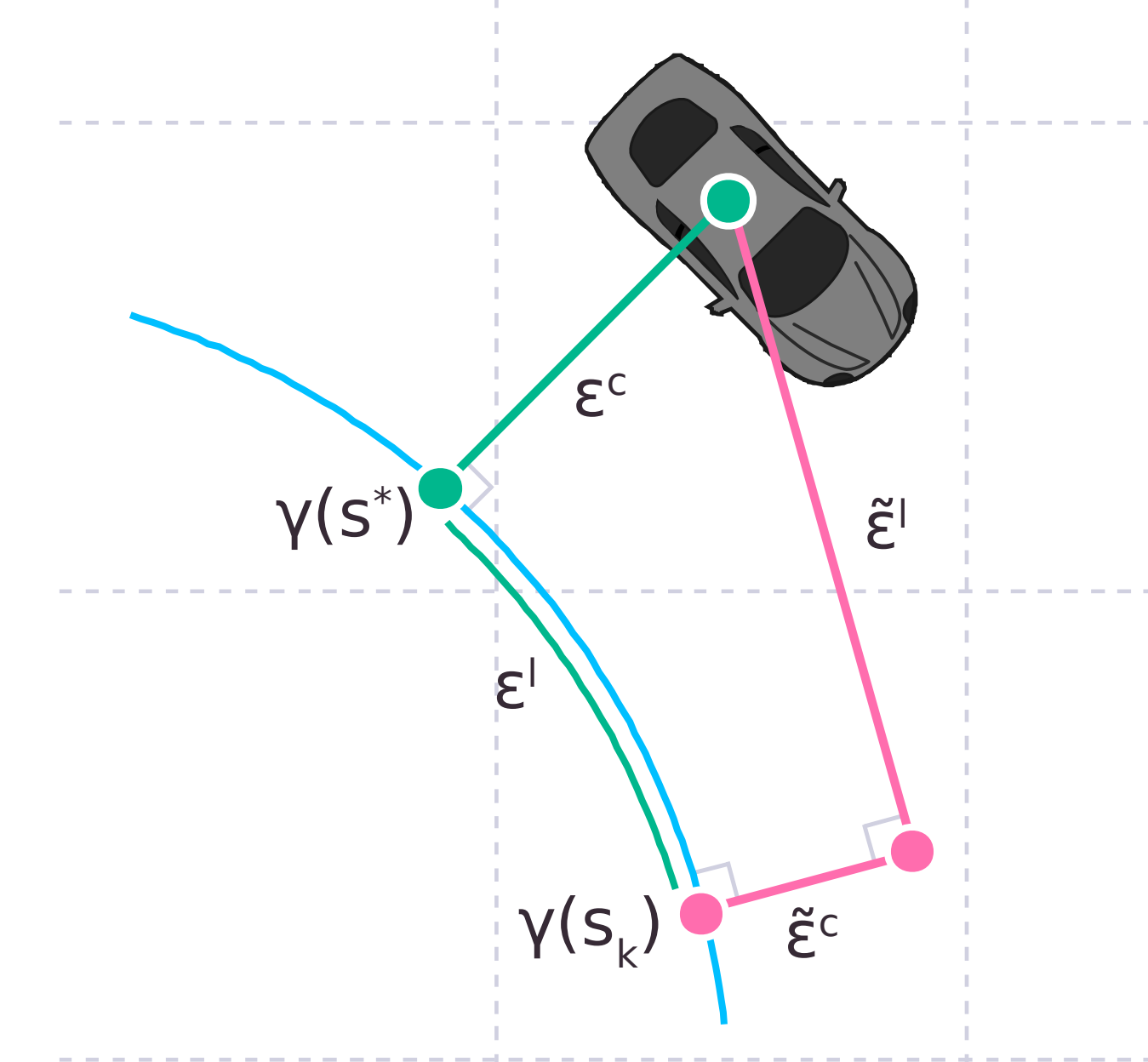
CARTPOLE SWINGUP



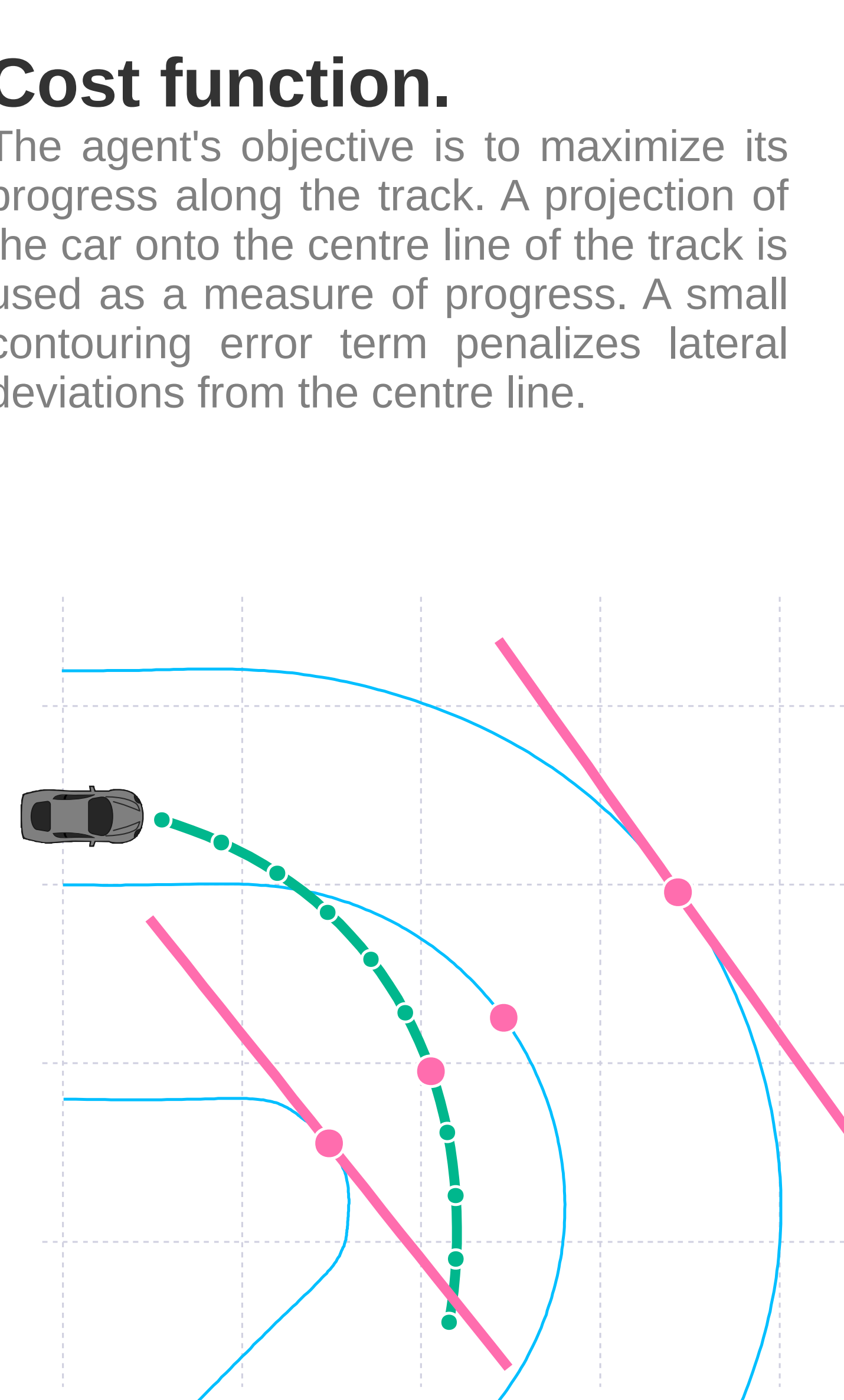
Constrained swingup.
The length of the track is limited; constraining the cart's position to between -2m and 2m. GP-RHC learns to solve the cartpole swing-up task in just a few episodes. Online updates improve learning rate, reduce variability in cost and result in fewer constraint violations.

Comparison with PILCO.
Since PILCO cannot handle constraints directly, the track limits are removed for comparison. GP-RHC is competitive with PILCO both in terms of sample efficiency and overall performance.

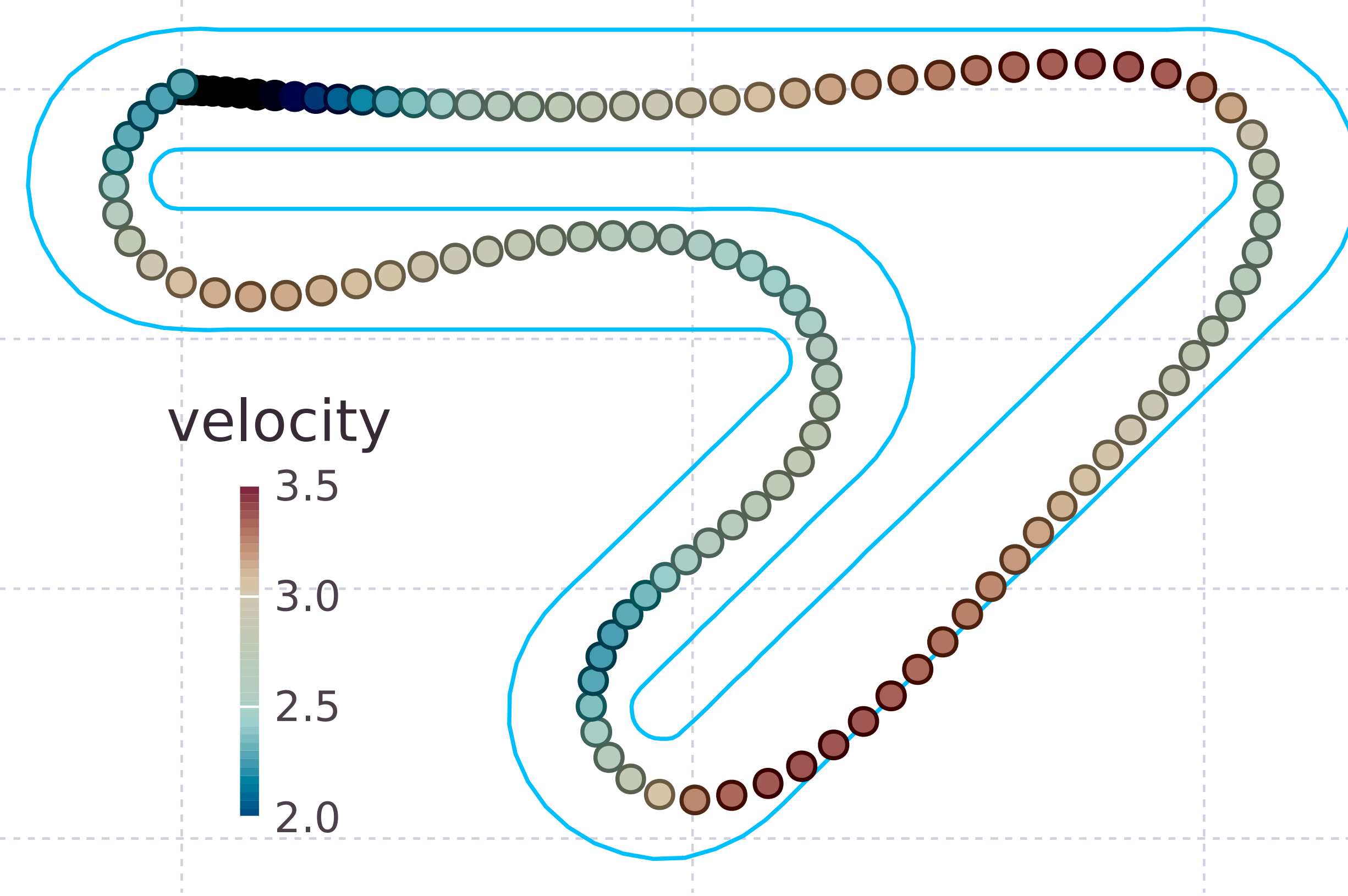
AUTONOMOUS RACING



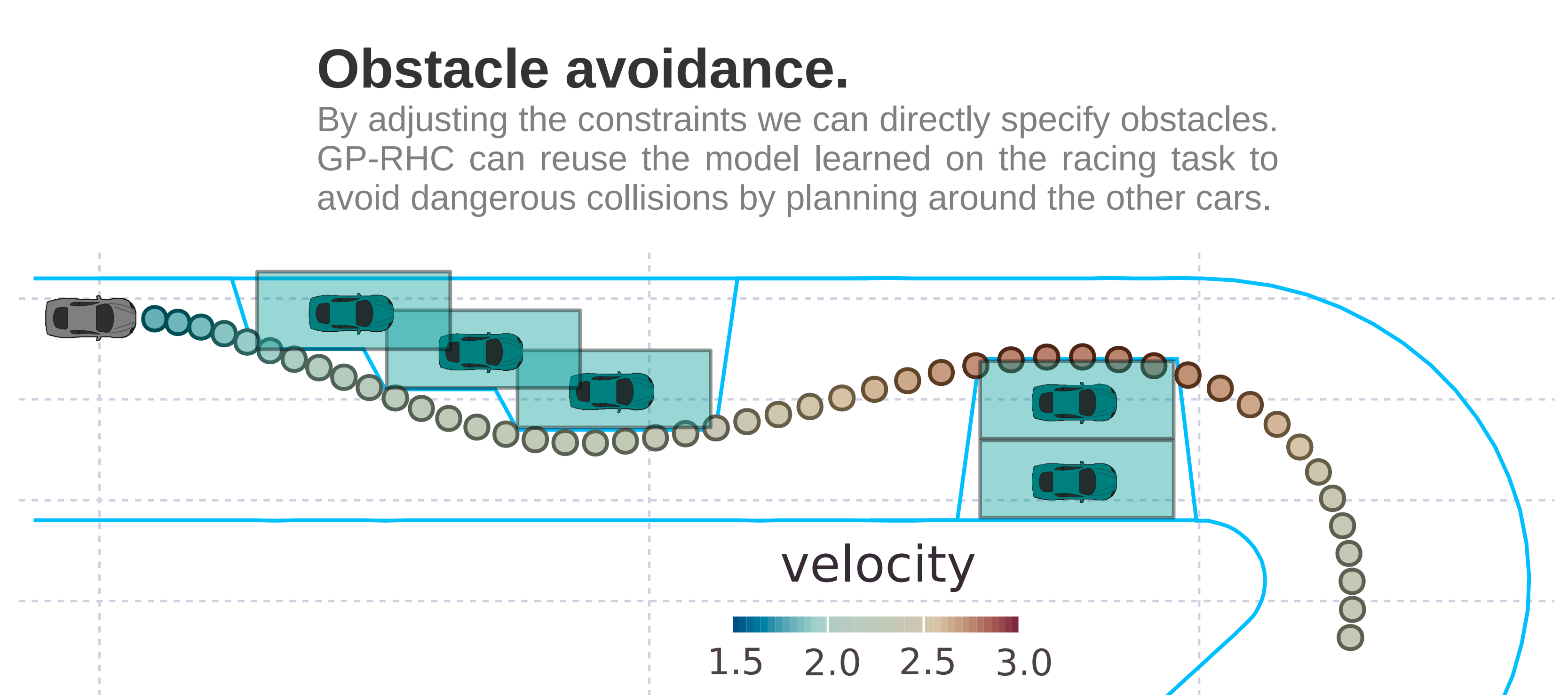
Track Constraints.
To ensure that the car remains within the track, limits are placed on the car's position. Each point on the planned trajectory is constrained to lie within two half spaces defined by the left and right track boundaries.



Cost function.
The agent's objective is to maximize its progress along the track. A projection of the car onto the centre line of the track is used as a measure of progress. A small contouring error term penalizes lateral deviations from the centre line.



Learning to race.
The dynamics model was initialized using 70 data points collected on a simpler oval track. Using the learned model, GP-RHC plans and drives trajectories that satisfy the complex track constraints. Despite limited training data the sparse spectrum GPs were able to generalize well enough to effectively navigate the sharp left turn.



Obstacle avoidance.
By adjusting the constraints we can directly specify obstacles. GP-RHC can reuse the model learned on the racing task to avoid dangerous collisions by planning around the other cars.

REFERENCES

- Alexander Domahidi and Juan Jerez. FORCES Professional. embotech GmbH (<http://embotech.com/FORCES-Pro>), July 2014.
- Miguel Lázaro-Gredilla, Joaquin Quiñero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. The Journal of Machine Learning Research, 2010.