Learning to Fail: Failure Plans and Predictions for Crowd Navigation

Patrick Naughton, Tushar Kusnur, Ishani Chatterjee, Maxim Likhachev

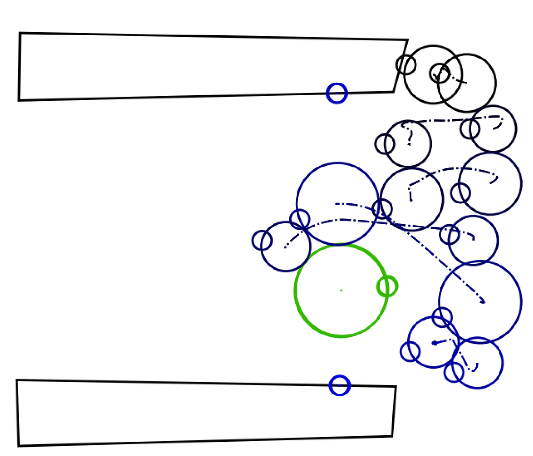
Robotics Institute, Carnegie Mellon University

Introduction

Objective: Create a controller to generate failure trajectories when a mobile robot fails to reach a waypoint. Additionally, make a predictor that identifies the distribution of next locations the robot could take given its state.

Motivation

- Robots navigating amongst crowds of people need to obey implicit human social rules and avoid many dynamic obstacles.
- For long term planning to distant goals, we seek to separate planning from controllers to handle small steps.
- Because these controllers may fail, we seek to develop a "failure controller" to guide the robot along a contingency trajectory.
- We would also like to predict this trajectory at planning time so the planner can perform state expansions from it.



Having the robot (green circle) remain still may leave it in an area heavily trafficked by humans (blue circles) causing collisions or inconveniencing the humans.

Approach

Failure Controller

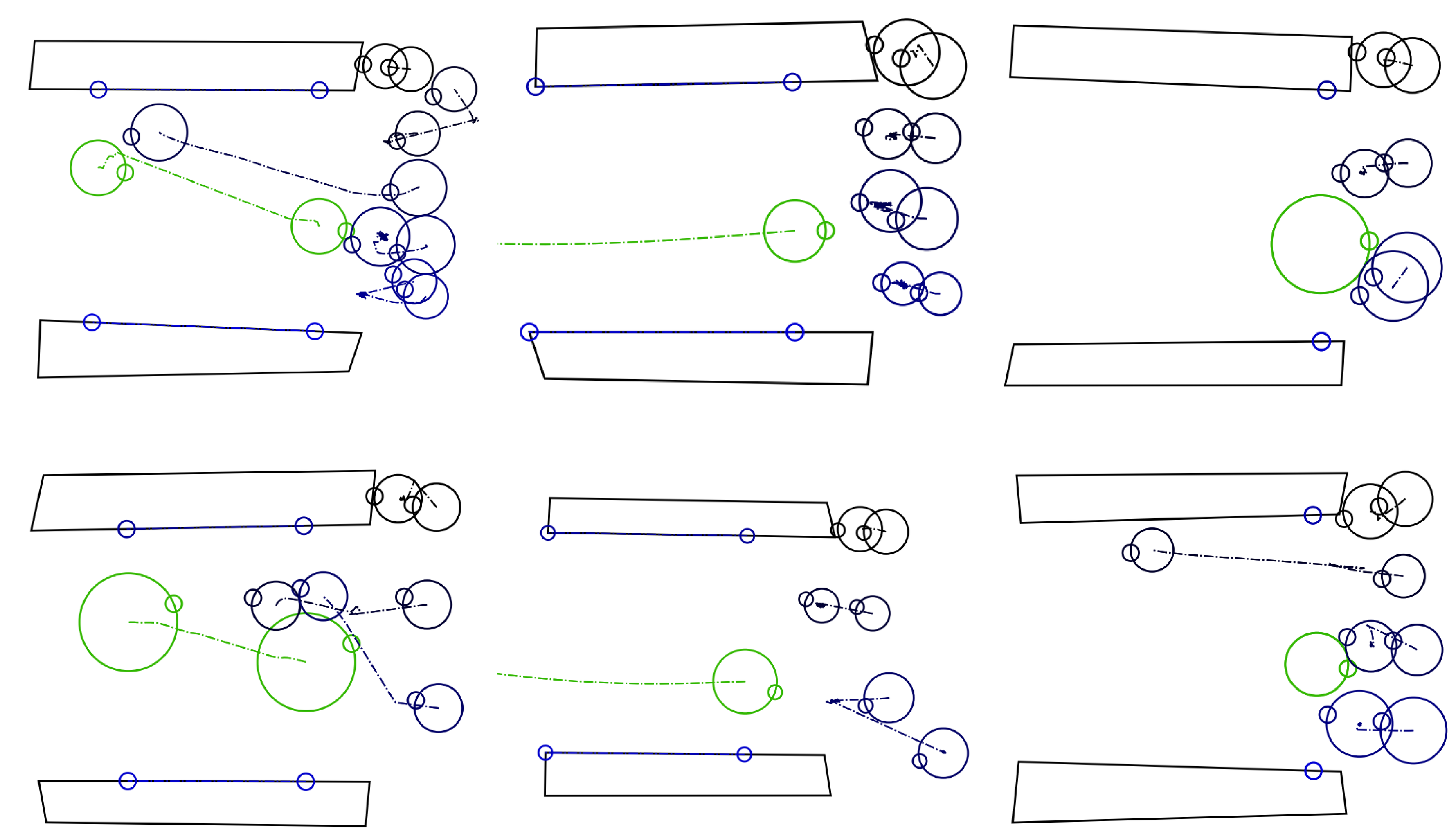
- Model navigation as a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ to achieve the maximum long term return.
- Use deep reinforcement learning (specifically the DQN algorithm [1]) to learn a function $Q(s_t, a_t)$ which gives the long term reward of action a_t from state s_t.
- Once we know Q, we can easily construct an optimal trajectory by always selecting action $argmax_a Q(s_t, a)$.

Failure State Prediction

- Simulate and record trajectories for the failure controller.
- Train a neural network to predict the distribution of next locations given the current state. We assume that the final distribution is Gaussian, so the network learns the parameters $\mu_x, \mu_y, \sigma_x, \sigma_y, \rho$ that characterize it.

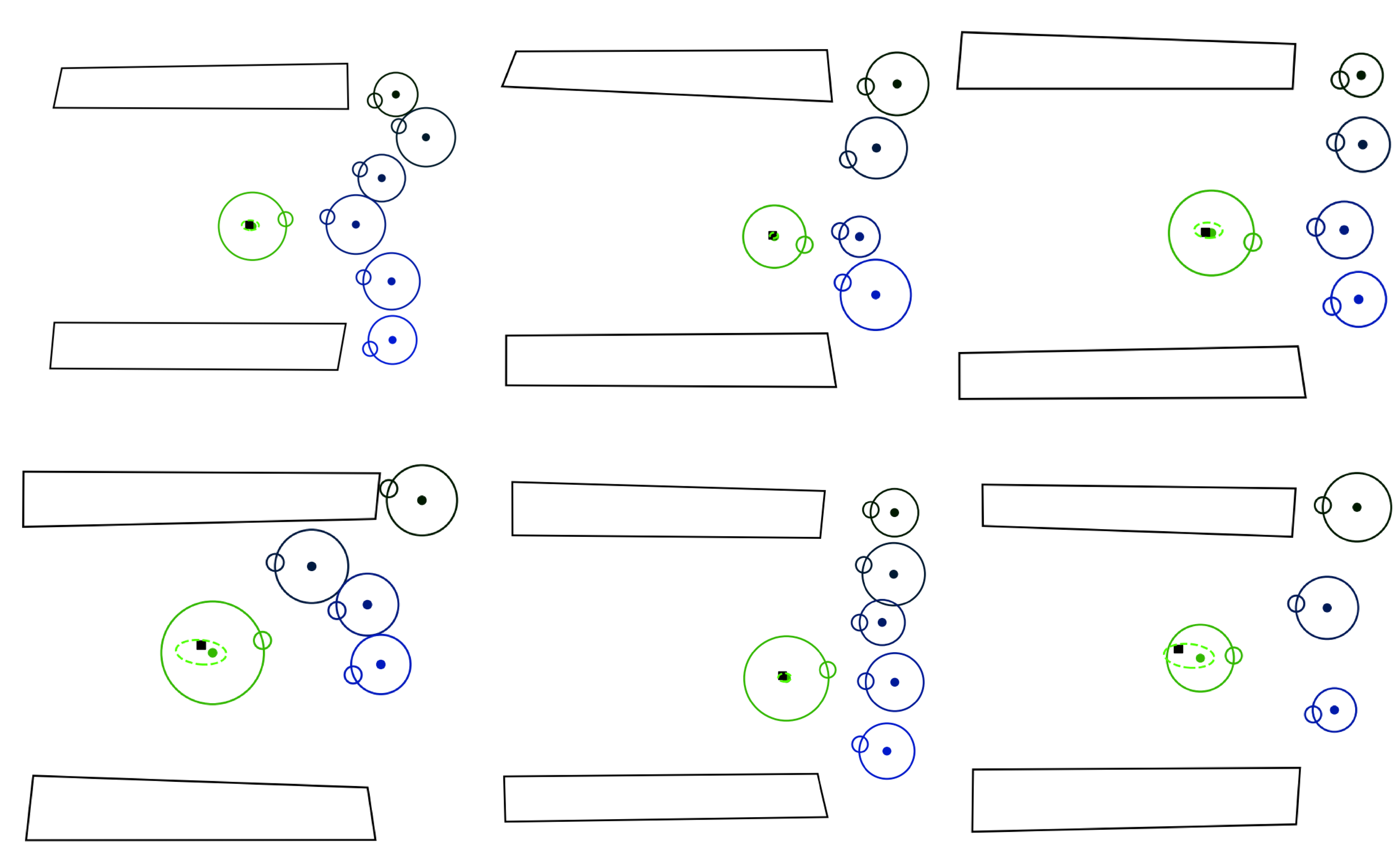
Failure Controller Generates Safe and Predictable Trajectories

Failure Controller



A comparison between our controller (left), a social forces controller (middle), and a do nothing controller (right). The initial and final locations of the robot and each human are shown, connected by a dotted line to illustrate their trajectories. The social forces trajectories continue to the left in an approximately straight line until they reach their time-out.

Failure Predictor



Examples of the predictor's output. At each time step, the failure predictor estimates the robot's next position and its uncertainty in that prediction. The dashed green ellipse shows the region in which the predictor is 99% confident the robot will be next and the black square shows the robot's actual next location. The top row shows results from training data which have much lower variance as expected.

Failure Controller Balances Safety and Efficiency

	RL (Ours)	Social Forces	Do Nothing
Path Length	1.23 (0.74)	14.70 (0.00)	0.07 (0.12)
Angular Distance	21.15(13.27)	4.18 (1.47)	10.28 (6.95)
Time	5.61 (2.96)	10.00 (0.00)	3.39 (3.45)
Collisions	0.02 (0.14)	0.00 (0.00)	0.75 (4.74)
Intrusions	34.72(47.11)	0.76 (2.41)	190.58(60.71)

Comparison between the presented controller, a social forces controller, and a do nothing controller. The mean of each metric over 100 runs is shown with the standard deviation in parentheses.

Conclusion and Future Work

This work presents a failure controller which successfully balances safety with path efficiency and a predictor to forecast its trajectories at planning time.

Future Work

- Apply this method to other controllers for different behaviors (for example, overtaking pedestrians).
- Develop a robust way to determine, at execution time, when to end the failure controller.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," CoRR, vol. abs/1312.5602, 2013.

Acknowledgements

- Thank you to Professor Maxim Likhachev as well as Ishani Chatterjee and Tushar Kusnur.
- Thank you to Rachel Burcin and Professor John Dolan.

THE ROBOTICS INSTITUTE

RISS ROBOTICS INSTITUTE SUMMER SCHOLARS