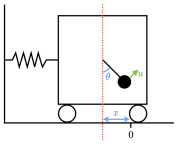
A Appendix

A.1 Benchmark Description

Adaptive Cruise Control. The first benchmark, Adaptive Cruise Control, involves an ego vehicle and a lead vehicle with 6 variables representing the position, velocity and acceleration of the two vehicles. Our training objective is to learn a linear controller that, when the lead vehicle suddenly reduces its speed, the ego car can decelerate to maintain a safe distance. φ_{safe} specifies the minimum relative distance that a controller should maintain at each timestep as well as an upper bound to prevent the ego car from stopping. A rollout lasts 50 timesteps and each timestep is $\delta = 0.1s$. VEL learned a safe controller because it uses verification feedback to directly optimize the worst-case safety loss in the proof space. Previous work [43] used verification to detect safety issues for a well-trained controller for this system. Our result shows that verification can also be used for synthesizing safe-by-construction controllers.

Oscillator. This is our runtime example in Sec. 1.

Tora. The Tora (translational oscillations by a rotational actuator) model (depicted on the right) is that of a cart attached to a wall with a spring, and is free to move on a friction-less surface. The cart itself has a weight attached to an arm inside it, which is free to rotate about an axis. This serves as the control input, in order to stabilize the cart at x = 0. The model is a 4 dimensional system, given by the following differential equations:



$$\dot{x_1} = x_2 \qquad \dot{x_2} = -x_1 + 0.1 \cdot sin(x_3)$$
$$\dot{x_3} = x_4 \qquad \dot{x_4} = a$$

The verification problem in the competition is that for an initial set of $x_1 \in [0.6, 0.7]$, $x_2 \in [-0.7, -0.6]$, $x_3 \in [-0.4, -0.3]$, and $x4 \in [0.5, 0.6]$, the system states stay within the box $x \in [-2, 2]^4$, for a bounded time window (20s). VEL can easily verify this property. To make the problem more interesting, we added a new specification for φ_{reach} as an inductive invariant, requiring that any rollout starting from φ_{reach} eventually turns back to it and any rollout states must be safe (including those that temporarily leave φ_{reach}).

Unicyclecar. The unicycle car benchmark can be expressed by the following dynamics equation:

$$\dot{x}_1 = x_4 \cos(x_3) \qquad \dot{x}_2 = x_4 \sin(x_3)$$
$$\dot{x}_3 = a_2 \qquad \dot{x}_4 = a_1 + w$$

where w is a random bounded error in the range $[-1e-4, 1e-4]$. In our setting, we set the sampling period $\delta = 0.05$ and the total time is $5s$ (100 control steps). The initial set is $[9.5, 9.55] \times [-4.5, -4.45] \times [2.1, 2.11] \times [1.5, 1.51]$. The goal set is $[-0.6, 0.6] \times [-0.2, 0.2] \times [-0.06, 0.06] \times [-0.3, 0.3]$.

Table 3: The system dynamics of our benchmarks.

Name	Dynamics	T	δ
B_1	$\dot{x}_1 = x_2, \dot{x}_2 = ux_2^2 - x_1$	35	0.2s
B_2	$\dot{x}_1 = x_2 - x_1^3, \dot{x}_2 = u$	9	0.2s
B_3	$\dot{x}_1 = -x_1(0.1 + (x_1 + x_2)^2), \dot{x}_2 = (u + x_1)(0.1 + (x_1 + x_2)^2)$	60	0.1s
B_4	$\dot{x}_1 = -x_1 + x_2 - x_3, \dot{x}_2 = -x_1(x_3 + 1) - x_2, \dot{x}_3 = -x_1 + u$	5	0.1s
B_5	$\dot{x}_1 = x_3^3 - x_2, \dot{x}_2 = x_3, \dot{x}_3 = u$	10	0.2s
$\text{Oscillator}_{\texttt{inf}}$	$\dot{x}_1 = x_2,\ \dot{x}_2 = (1 - x_1^2)x_2 - x_1 + u$	150	0.01s
ACC	$\dot{x}_1 = x_2,\ \dot{x}_2 = x_3,\ \dot{x}_3 = -4 - 2x_3 - 0.0001x_2^2$ $\dot{x}_4 = x_5,\ \dot{x}_5 = x_6,\ \dot{x}_6 = 2u - 2x_6 - 0.0001x_5^2$	50	0.1s
MountainCar	$x_1^+ = x_1 + x_2\ x_2^+ = x_2 + 0.0015u - 0.0025\cos(3x_1)$	150	-
QMPC	First Segment: $\dot{x}_1 = x_4 - 0.25,\ \dot{x}_2 = x_5 + 0.25,\ \dot{x}_3 = x_6,$ $\dot{x}_4 = 9.81\tan(u_1),\ \dot{x}_5 = -9.81\tan(u_2),\ \dot{x}_6 = u_3 - 9.81$	30	0.2s
$\text{Pendulum}_{\texttt{inf}}$	$\dot{x}_1 = x_2, \dot{x}_2 = 15\sin(x_1) + 3u$	15	0.05s
CartPole	$\dot{x}_1 = x_2, \dot{x}_2 = t - tc \cdot \cos(x_3)/22$ $\dot{x}_3 = x_4, \dot{x}_4 = tc,$ where $t = (20u + x_4^2\sin(x_3))/22$ and $tc = (9.8\sin(x_3) - t \cdot \cos(x_3))/(2/3 - \cos(x_3^2)/20)$	50	0.02s
UnicycleCar	$\dot{x}_1 = x_4\cos(x_3),\ \dot{x}_2 = x_4\sin(x_3),\ \dot{x}_3 = u_2,\ \dot{x}_4 = u_1 +$ w, $(w \in [1e-4, 1e-4])$	30	0.2s
Tora	$\dot{x}_1 = x_2, \dot{x}_2 = -x_1 + 0.1\sin(x_3),\ \dot{x}_3 = x_4, \dot{x}_4 = u$	10	0.5s
$\text{Tora}_{\texttt{inf}}$	$\dot{x}_1 = x_2, \dot{x}_2 = -x_1 + 0.1\sin(x_3),\ \dot{x}_3 = x_4, \dot{x}_4 = u$	16	0.1s

On all the benchmarks, the reinforcement learning algorithm [38] can obtain high reward controllers. However, these controllers cannot be verified safe due to approximation erros. VEL optimizes these controllers on top of the proof space and generates verifably safe controllers very efficiently.