CACTO: Continuous Actor-Critic algorithm with Trajectory Optimization towards global optimality

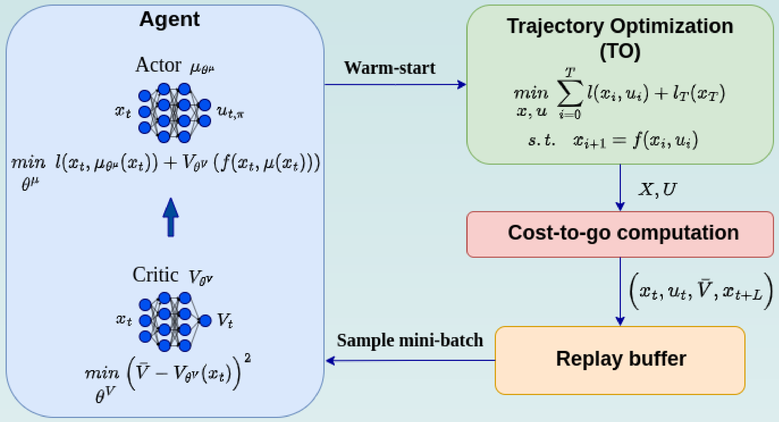
Gianluigi Grandesso*, Gastone P. Rosati Papini*, Patrick M. Wensing** and Andrea Del Prete*

*University of Trento, Department of Industrial Engineering
**University of Notre Dame, Department of Aerospace and Mechanical Engineering

Abstract

This poster presents a novel algorithm for the continuous control of dynamical systems that combines **Trajectory Optimization (TO)** and **Reinforcement Learning (RL)** in a single framework. The motivations behind this algorithm are the two main limitations of TO and RL when applied to continuous systems to minimize a **non-convex cost function**. Specifically, TO can get stuck in poor local minima when the search is not initialized close to a "good" minimum. On the other hand, when dealing with continuous state and control spaces, the RL training process may be excessively long and strongly dependent on the exploration strategy adopted and its sample efficiency. Thus, our algorithm aims to **learn a "good" control policy via RL policy search guided by TO**, that, when used as **initial guess provider for TO**, makes the trajectory optimization process less prone to converge to poor local optima.

Schematic



- Actor and Critic approximated by Deep Neural Networks (DNN)
- Critic updated to match the computed cost-to-go
- Actor updated by following the Critic's gradient
- Trajectory Optimization problem warm-started with Actor's policy rollout
- State-action space explored by performing TO control trajectories

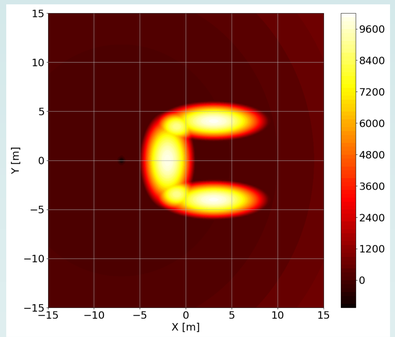
Algorithm

Inputs: dynamics $f(\cdot,\cdot)$, running cost $l(\cdot,\cdot)$, terminal cost $l_T(\cdot)$, T, M, S, K, L
Output: trained control policy $\mu(x)$
$\theta^V \leftarrow$ random, $\theta^\mu \leftarrow$ random, $\theta^{V'} \leftarrow \theta^V$
Initialize replay buffer R
for $episode \leftarrow 1$ **to** M **do**
  $x_0 \leftarrow$ random, $x^{init}_{TO,0} \leftarrow x_0$
  for $t \leftarrow x_0[n]$ **to** T **do** *(policy rollout)*
    $u^{init}_{\mu,t} \leftarrow \mu(x^{init}_{\mu,t} | \theta^\mu)$
    $x^{init}_{\mu,t+1} \leftarrow Environment(x^{init}_{\mu,t}, u^{init}_{\mu,t})$
  end
  $(x^{init}_{TO}, u^{init}_{TO}) \leftarrow (x^{init}_{\mu}, u^{init}_{\mu})$ *(TO warm-start)*
  Solve TO problem and get control trajectory u_{TO}
  Agent's initial state $\leftarrow x_0$
  for $t \leftarrow x_0[n]$ **to** T **do** *(episode rollout)*
    $x_{t+1}, l_t \leftarrow Environment(x_t, u_{TO,t})$
  end
  for $t \leftarrow x_0[n]$ **to** T **do**
    Compute partial cost-to-go:
    $\hat{V}_t = \sum_{j=0}^{L} (1 - \mathbb{1}_A(t,j))\; l_{t+j}, \quad A = \{t,j \mid t+j \geq T\}$
    $R \leftarrow (x_t, u_{TO,t}, \hat{V}_t, x_{min(t+L+1,T)})$
  end
  if $episode \;\%\; e_{update} = 0$ **then**
    for $k \leftarrow 1$ **to** K **do** *(critic & actor update)*
      Sample minibatch of S transitions $(x_i, u_{TO,i}, \hat{V}_i, x_{min(i+L+1,T)})$
      Compute cost-to-go: $\bar{V}_i = \hat{V}_i + (1 - \mathbb{1}_A(i+1, L))\, V'(x_{i+L+1})$
      Update critic by minimizing (over θ^V) the loss: $L_c = \frac{1}{S} \sum_{i=1}^{S} \left(\bar{V}_i - V(x_i)\right)^2$
      Update actor by minimizing (over θ^μ) the loss: $L_a = \frac{1}{S} \sum_{i=1}^{S} Q(x_i, \mu(x_i))$
      Update target critic: $\theta^{V'} \leftarrow \tau\theta^{V'} + (1-\tau)\,\theta^V$
    end
  end
end

Results

- Task: find the **shortest path** to a target point while ensuring **low control effort** and **avoiding an obstacle**

Cost Function



- Tested on four systems: 2D point with single integrator dynamics, 2D point with double integrator dynamics, Dubins car, 3-DoF planar manipulator

Comparison w.r.t. random initial guess and ICS as warmstart

System	x,y ∈ [-15,15]m		x ∈ [0,15]m y ∈ [-5,5]m	
	CACTO vs. Random	CACTO vs. ICS	CACTO vs. Random	CACTO vs. ICS
2D Point Single Int.	99.88% (99.88%)	14.49% (99.88%	99.11% (99.11%)	91.96% (99.11%)
2D Point Double Int.	99.88% (99.88%)	12.38% (99.88%)	99.88% (99.88%)	91.96% (99.11%)
Dubins Car	89.72% (98.83%)	15.65% (95.56%)	99.88% (99.88%)	92.86% (100%)
Manipulator	91.78% (91.91%)	77.94% (78.33%)	99.88% (99.88%)	100% (100%)

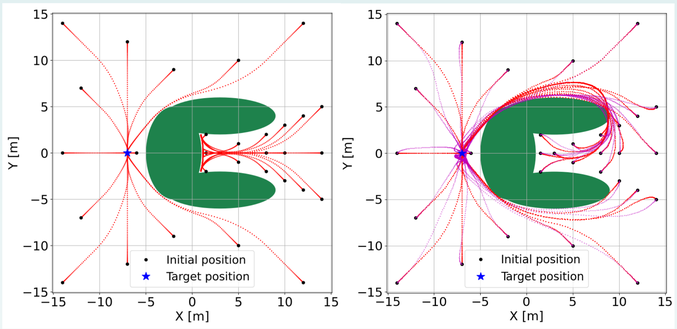
Conclusion

- In all our tests, using rollouts of the policy learned by CACTO outperforms standard warm-starting techniques, such as randomizing the initial guess or choosing the initial conditions (ICS), in terms of cost
- Considering a discrete-space version of CACTO using look-up tables instead of DNN, we proved that the policy can only improve as the algorithm proceeds
- Besides being used as initial guess provider for TO, it can also be used as a model- based deep RL algorithm to find directly the control policy

Future work

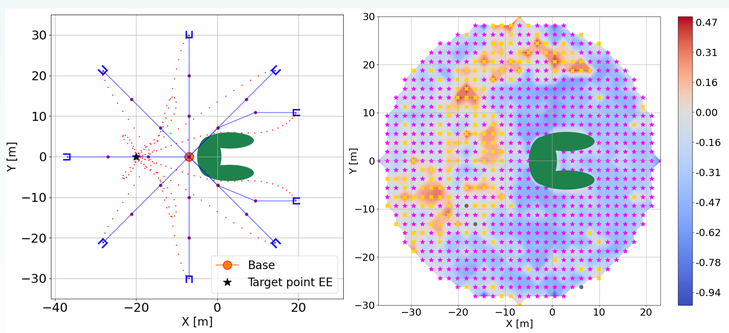
- Application to higher dimensional systems and different context
- Training acceleration through the implementation of Sobolev Learning
- Extension to Co-Design by including also HW optimization

2D point with double integrator dynamics



ICS warmstart *CACTO warmstart*

3-DoF planar manipulator



TO trajectories, ICS warmstart *Normalized cost difference*