Robot Learning
Weekly Exercise 10

Marc Toussaint & Wolfgang Hönig

Learning & Intelligent Systems Lab, Intelligent Multi-Robot Coordination Lab, TU Berlin

Marchstr. 23, 10587 Berlin, Germany

Summer 2024

1 Literature: Learning to Plan in TAMP

Here is an example paper for learning to plan:

D. Driess, J.-S. Ha, and M. Toussaint. Deep Visual Reasoning: Learning to Predict Action Sequences for Task and Motion Planning from an Initial Scene Image, 2020-06-09. URL: http://arxiv.org/abs/2006.05398, arXiv: 2006.05398

The paper trains an image-based action sequence prediction. A follow-up paper[1] does something similar with a much more ambitious Large Manguage Model, but the above paper more clearly defines the problem in relation to TAMP. To get an overview, you may first watch the video https://www.youtube.com/watch?v=i8yyEbbvoEk.

Here are the questions:

a) Eq. (4) defines the action sequence prediction model π. Note that S is the scene, g the goal, and $a_{1:K} \in \mathcal{T}(g, S), F_S(a_{1:K}) = 1$ means "$a_{1:K}$ is feasible and leads to goal g".

How does this π relate to modern sequence/language models, which also predict the next word/token given previous tokens? (What exactly is similar and different?)

How does this π relate to a trained state evaluation function as they are used, e.g., in modern chess/go engines? (Which score a board and therefore provide a heuristic for search. What exactly is similar and different?)

The analogy is: $a_{1:k\text{-}1}$ are tokens of previous words. Similar: It predicts the next word (action). Different: Actually it predicts the "probability of the next word", or rather that the next word is part of a feasible sentence. Also unusual in classical LLMs: This prediction is conditional to other information: The scene and the goal. That's similar to modern "multi-modal" LLMs, which can take words, images, anything as input (e.g. PaLM-E).

π is exactly the 'optimal' state-evaluation function in a one-player game, where the return is binary, indicating feasibility of $a_{1:K}$. 'optimal' in the sense of assuming optimal continuation of future decisions (as in Bellman optimality) – which is why we have the $\exists$ quantifiers. However, this state-evaluation function is conditional to S, g, and the state is the full $a_{1:k\text{-}1}$ so far. (See section on "Relation to Q-Function").

b) In Eq. (4), the actions a_k are input to the network. But they are encoded in a very particular way, as explained in subsection C (see also video at 0:20sec). How exactly are actions encoded?

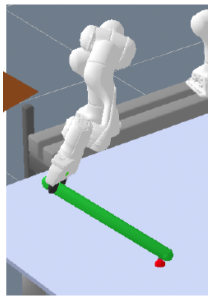
Image (depth) paired with object masks. Conveninet: There is no need to given them IDs or numbers. The mask has a universal format to "refer" to objects, no matter how many objects there are in the scene.

c) As always, understanding the data generation is key. Section V.B (page 7) explains the data generation process, and Eq. (5) the definition of D_{data} (ingnore D_{train}). In this whole process, how often was the feasibility $F_S(a_{1:K})$ of an action sequence $a_{1:K}$ in a scene S being computed. (This computation happended fully model-based assuming full knowledge of the scene instantiated in the simulator.)

[1] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. PaLM-E: An Embodied Multimodal Language Model, 2023-03-06. URL: http://arxiv.org/abs/2303.03378, arXiv:2303.03378

$102\,566 + 2\,741\,573 = 2\,844\,139$

2 Optimal Sequential Manipulation in TAMP



Consider the scene on the right, where we have one robot with 7 degrees of freedom (dofs) $q \in \mathbb{R}^7$, and a stick with its pose $s \in \mathrm{SE}(3)$ as degrees of freedom. (Ignore the triangle in the image.)
As discussed in the lecture, we consider the whole scene as a single multibody system with (q, s) as its configuration. Initially the stick is lying somewhere on the table (away from the red ball); the final goal is for the stick to touch the red ball.

Assume that you have access to three constraint functions:

- $\phi_{\text{grasp}}(q, s) \in \mathbb{R}^3$ is a 3-dimensional constraint such that $\phi_{\text{grasp}}(q, s) = 0$ indicates a correct (stable) grasp of the stick by the robot.
- $\phi_{\text{touch}}(s) \in \mathbb{R}^1$ is a 1-dimensional constraint such that $\phi_{\text{touch}}(s) = 0$ indicates that the stick touches the red ball.
- $\phi_{\text{coll}}(q, s) \in \mathbb{R}^1$ is a 1-dimensional constraint such that $\phi_{\text{coll}}(q, s) \le 0$ indicates that nothing in the scene collides.

a) Formulate a mathematical program that represents the problem of optimally grasping the stick and then, with the grasped stick, optimally touching the red ball. The problem should only be about finding the grasp pose and the final pose – not yet the motions in between. As usual, optimality should reflect minimal motion effort by the robot. Assume the initial configuration is $(q_0, s_0) \in \mathbb{R}^7 \times \mathrm{SE}(3)$.

$$\min_{q_{1,2}, s_{1,2}} \sum_{t=1}^{2} \|q_t - q_{t\text{-}1}\|^2 + \|s_t - s_{t\text{-}1}\|^2 \tag{1}$$

$$\text{s.t.}\quad \forall_{t=1,2} : \phi_{\text{coll}}(q_t, s_t) = 0 \tag{2}$$

$$\phi_{\text{grasp}}(q_1, s_1) = 0 \tag{3}$$

$$\phi_{\text{touch}}(s_2) = 0 \tag{4}$$

$$s_1 = s_0 \tag{5}$$

$$\mathrm{relPose}(s_2, q_2) = \mathrm{relPose}(s_1, q_1) \tag{6}$$

The last line is really the caveat! Imposing that the relative pose of the stick in hand is the same in time slice 1 and 2.

b) It is quite natural to choose (q_1, s_1, q_2, s_2) as the decision variables of the above mathematical program. But can you think of an alternative, lower-dimensional parameterization of the problem?

$s_1 = s_0$ so we don't need that. And we introduce $r = \mathrm{relPose}(s_2, q_2)$ as the actual decision variable, and $s_t = s(r, q_t)$ as a direct function of the robot pose and the grasp pose. Esp. if the manipulation sequence was longer, this is much lower dimensional.

$$\min_{q_{1,2}, r} \sum_{t=1}^{2} \|q_t - q_{t\text{-}1}\|^2 \tag{7}$$

$$\text{s.t.}\quad \forall_{t=1,2} : \phi_{\text{coll}}(q_t, s(r, q_t)) = 0 \tag{8}$$

$$\phi_{\text{grasp}}(q_1, s_0) = 0 \tag{9}$$

$$\phi_{\text{touch}}(s(r, q_2)) = 0 \tag{10}$$

$$s(r, q_1) = s_0 \tag{11}$$

c) Now modify the mathematical program above (of a) or b)) to include the full motion from the start configuration until the stick touches the ball. Use an optimality criterion as is standard in trajectory optimization.

$$\min_{q_{[0,2]}, r} \int_{t=0}^{2} \|\ddot q(t)\|^2 \tag{12}$$

$$\text{s.t.} \quad q(0) = \dot q(0) = 0 \tag{13}$$

$$\forall_{t\in[0,1]} : \phi_{\text{coll}}(q(t), s_0) = 0 \tag{14}$$

$$\forall_{t\in[1,2]} : \phi_{\text{coll}}(q(t), s(r, q(t))) = 0 \tag{15}$$

$$\phi_{\text{grasp}}(q(1), s_0) = 0 \tag{16}$$

$$\phi_{\text{touch}}(s(r, q(2))) = 0 \tag{17}$$

$$s(r, q(1)) = s_0 \tag{18}$$

d) Neglect the motion again; consider only grasp and touch. But this time consider a sequence of 4 actions: grasp-stick, place-stick, grasp-stick, touch-ball, where the 2nd action places the stick back on the table before picking it up again. Can you think of scene (stick and ball placement) where this action sequence makes sense? Instead of $(q_1, s_1, q_2, s_2, q_3, s_3, q_4, s_4)$, what would be a lower-dimensional parameterization?

If the ball is very far, so that the stick needs to be grasped at an end; but the stick is also far and can only grasped in the middle; then a re-grasp of the stick is necessary. The optimization formulation above can solve for this.

First $s_1 = s_0$ and $s_3 = s_2$, as it is resting on the table. We can also replace s_2 by a relative placement parameter $p = \text{relPose}(s_2, \text{table})$, which is only 3 or 4dof as the stick needs to lay flat.

(For discussion at the tutorial:) You know how path finding in a standard setting is defined as finding a collision free path.[2] How can the same sequential manipulation problem as in b) be represented as a path finding problem (respecting all constraints but neglecting optimality)?

Multi-modal motion planning – but one runs into so many issues. Naive/random search over mode transitions/switches can kill efficiency.

References

[1] D. Driess, J.-S. Ha, and M. Toussaint. Deep Visual Reasoning: Learning to Predict Action Sequences for Task and Motion Planning from an Initial Scene Image, 2020-06-09. URL: http://arxiv.org/abs/2006.05398, arXiv:2006.05398.

[2] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. PaLM-E: An Embodied Multimodal Language Model, 2023-03-06. URL: http://arxiv.org/abs/2303.03378, arXiv:2303.03378.

[2] E.g., finding a continuous path $\gamma : [0, T] \to \mathcal{X}_{\text{free}}$ from a given start configuration $\gamma(0) = x_0$ to a final configuration $\gamma(T) \in \mathcal{X}_{\text{goal}}$ within the free configuration space $\mathcal{X}_{\text{free}} = \{x \in \mathcal{X} : \phi_{\text{coll}} \le 0\}$.