PhD Proposal:
Combined Task and Motion Planing for Multi Robotic Arms in Pick/Place Problems

By: Nir Levi

November 8, 2015

1 Introduction

Imagine how life would be like if evolution would created us with another arm. House holding tasks such as cooking, dish washing or baby care would preform much easier and more faster. Imagine a guitar, drums or piano players (or any other musicians) enjoying three arms, what reacher music we could have listen to. Imagine a surgeon a soldier a construction or fruit peaking worker or even us using a computer with an additional hand. These all are simple examples of advanced capabilities obtained by adding only a single arm to our body, now think of adding 10.

2 Literature Review

[1] **Decentralized path planning for multi-agent teams with complex constraints**

This paper discuss the challenge of planning paths for multi-agent systems. The paper shows an improved Closed-Loop RRT motion planner that handle multi agent system based on dynamic priority of an individual agent plan.

[2] Fast Distributed Multi-Agent Plan Execution with Dynamic Task Assignment and Scheduling

This article shows the same case study as ours but with a different point of view. The article discuss on execution of a plan and the consequences of a lack of accuracy during execution. This paper introduces Chaski, a multi-agent executive for scheduling temporal plans with online task assignment.

3 Methodology

In this chapter we will go through an application of methods described in the literature review for development of task planning techniques involving multi robotic arms. First we describe the problem we will contend with and present a simple case study example for which it will be easy to examine various courses of solutions.

3.1 Problem Description

Given the following input data:

- Initial and goal positions of a block b_{init}, b_{goal}.
- A workspace of N robotic arms $ws = \{w_1, w_2, ..., w_N\}$

What is the workspace sequence $wss = w_{init}, ..., w_k, w_{k+1}, ..., w_{goal}$ which will satisfy the next description:

- $b_{init} \in w_{init}$ and $b_g \in w_{goal}$ where $w_{init}, w_{goal} \in ws$
- $w_k \cap w_{k+1} \neq \emptyset$
- Having wss what is the path $p_k \in w_k$ of the k arm which makes no collision while moving the box through its workspace.

3.2 Algorithm

Algorithm 1 Main Loop

input: $b_{init}, b_{goal}, w_1, w_2, ..., w_N, Q^i_{init}$
output: $w_{init}, ..., w_k, w_{k+1}, ..., w_{goal}$
$t_{init}, ..., t_k, t_{k+1}, ..., t_{goal}$
procedure:
$w_{goal} \leftarrow \{w_n | b_{goal} \in w_n, n \in [1, N]\}$
$w_{init} \leftarrow \{w_n | b_{init} \in w_n, n \in [1, N]\}$
if $\left(w_{init} = \emptyset\right) \cup \left(w_{init} = \emptyset\right)$ **than** **return** No Solution
$cws \leftarrow$ **FindAllCoupledWS** $(w_1..w_N)$ // set of coupled WS
$wss \leftarrow$ **GenerateSequences** $(w_{init}, w_{goal}, cws)$ // set of WS sequences
if $\left(wss = \emptyset\right)$ **than** **return** No Solution
$wss \leftarrow$ **SortBySequenceLength** (wss)
for i=1..length(wss)
  $T \leftarrow$ **CalcSequnceTrajectory**$(wss\,(i)\,, b_{init}, b_{gaol})$
  if $T \neq \emptyset$
    execute(T)
    return success
return No Solution

Algorithm 2 Find all Shared Workspace

procedure: $FindAllCoupledWS\,(w_1..w_N)$
if (N==1) **than** res$\leftarrow \emptyset$
elseif $(N == 2) \cap (w_1 \cap w_2 \neq \emptyset)$ **than** $res \leftarrow (w_1 \cap w_2)$
elseif $(N == 2)$ **than** res$\leftarrow \emptyset$
else
  $s1 \leftarrow FindAllCoupledWS\,(w_2..w_N)$
  $s2 \leftarrow \{w_2..w_N, s1\}$
  res $\leftarrow \emptyset$
  for i = 1..length(s2)
    if $(w_1 \cap s2_i \neq \emptyset)$ **than** res$\leftarrow \{res, w_1 \cap s2_i\}$
  res$\leftarrow \{res, s1\}$
return res

Algorithm 3 Calculate Sequence trajectory

procedure: $CalcSequnceTrajectory(wss, b_{init}, b_{gaol}, Q^i_{init})$
$pkp \leftarrow b_{init}$ //pick position
$T \leftarrow \emptyset$ // Initialize Trajectory
for i=1..length(wss)
  if $i \neq length\,(wss)$
    $plp \leftarrow$ **FindPlacePosition** $(wss\,(i)\,, wss\,(i+1))$
  else $plp \leftarrow b_{goal}$
  $stp \leftarrow q^i_{init}$ // start position
  $T_1 \leftarrow$ **ClacTrajectory** (ws, stp, pkp)
  $T_2 \leftarrow$ **ClacTrajectory** (ws, pkp, plp)
  if $\left(T_1 = \emptyset\right) \cup \left(T_2 = \emptyset\right)$ **than** **return** $\emptyset$
  else $T \leftarrow \{T, T_1, T_2\}$
  $pkp \leftarrow plp$
return T

4 Preliminary Results

References

[1] V. Desaraju and J. How, "Decentralized path planning for multi-agent teams with complex constraints," *Autonomous Robots*, vol. 32, no. 4, pp. 385–403, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9275-2

[2] J. A. Shah, P. R. Conrad, and B. C. Williams, "Fast distributed multi-agent plan execution with dynamic task assignment and scheduling," in *Nineteenth International Conference on Automated Planning and Scheduling*, 2009.

[3] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 1930–1936. [Online]. Available: http://ijcai.org/papers15/Abstracts/IJCAI15-274.html

[4] C. Boutilier and R. I. Brafman, "Partial Order Planning with Concurrent Interacting Actions," *Artificial Intelligence*, vol. 14, pp. 105–136, 2001.

[5] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning." *I. J. Robotic Res.*, vol. 28, no. 1, pp. 104–126, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijrr/ijrr28.html#CambonAG09

[6] S. Cambon, F. Gravot, and R. Alami, "A robot task planner that merges symbolic and geometric reasoning," in *ECAI*, vol. 16, 2004, p. 895.

[7] R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," in *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, ser. IJCAI'71. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1971, pp. 608–620. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622876.1622939

[8] M. Fox and D. Long, "Pddl2.1: An extension to pddl for expressing temporal planning domains," *J. Artif. Int. Res.*, vol. 20, no. 1, pp. 61–124, Dec. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622452.1622454

[9] F. Lagriffoul, D. Dimitrov, J. Bidot, A. Saffiotti, and L. Karlsson, "Efficiently combining task and motion planning using geometric constraints," *I. J. Robotic Res.*, vol. 33, no. 14, pp. 1726–1747, 2014. [Online]. Available: http://dx.doi.org/10.1177/0278364914545811

[10] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. [Online]. Available: http://robotics.eecs.berkeley.edu/pubs/25.html