On-line robot execution monitoring using probabilistic action duration

(Extended Abstract)

V.A. Ziparo, L. Iocchi, M. Leonetti, D. Nardi
Dipartimento di Informatica e Sistemistica
via Ariosto 25, 00185
Rome, Italy
{lastname}@dis.uniroma1.it

ABSTRACT

The execution of tasks for a mobile robot embedded in a dynamic environment brings about several challenges, due to the dynamic changes of the environment and the inaccurate perception of the robot. This paper tackles the problem of on-line execution monitoring when the agent has different tasks and several plan to accomplish them, as in the BDI framework. Our method considers uncertainty in the duration of actions with a probabilistic model of action duration, and evaluates the cost of each possible plan at run-time in terms of probability of successful termination within a desired expected time.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Robot planning, Plan Execution Monitoring

1. INTRODUCTION

Robot plans are often simple, but rather tricky to formalize in terms of conditions and state properties that must be verified by the perceptive system. The relationship among planning, reasoning about actions and robot control is defined in the so-called *execution monitoring* component of a deliberative architecture [4, 3].

We propose an approach for execution monitoring, which takes into account the duration of actions in order to properly evaluate the chance of a plan to terminate successfully within a given time limit. The evaluation of the probability of successful plan execution is based on the on-line estimation of actions' duration based on the robot perception of the environment.

Although there has been some work in modeling durative actions with continuous change [2], it assumes a deterministic duration of actions. Given the uncertainty associated with the duration of robot complex actions, we argue that this aspect should be modeled by a probabilistic representation. Some work regarding probabilistic time models and computation of a probability density function for plan durations has been developed in the framework of MDPs [6, 5], but the computation is carried out off-line and there is no execution monitoring process. The model we propose does not require a specific action representation, nor an ad hoc planning approach; rather, it is inspired by a BDI (Belief Desires Intentions) architecture [1], where a set of plans are available for on-line plan selection.

Cite as: On-line robot execution monitoring using probabilistic action duration (Extended Abstract), V.A. Ziparo, L. Iocchi, M. Leonetti, D. Nardi, , *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1521-1522

2. PROBABILISTIC REPRESENTATION OF ACTION DURATION

The duration of an action α is denoted by a χ^2 probability distribution starting at time t_0 and with σ degrees of freedom. Its probability density function (*pdf*) is:

$$\chi^2(t-t_0|\sigma) = \begin{cases} \frac{1}{2^{\sigma/2}\Gamma(\sigma/2)} x^{(\sigma/2)-1} e^{(t_0-t)/2} & \text{for } t > t_0 \\ 0 & \text{for } t \leq t_0 \end{cases} \tag{1}$$

where $\Gamma(\cdot)$ denotes the Gamma function. Figure 1 shows some examples of χ^2 *pdf*s.

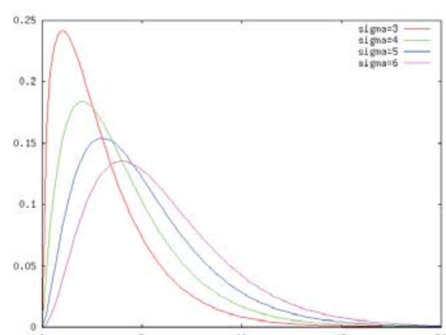

Figure 1: Examples of χ^2 probability density functions, when $t_0 = 0$ and σ varies from 3 to 6.

The motivation underlying the choice of the χ^2 distribution for modeling action durations is manifold:

- it can model the fact that the execution of an action for a robot requires at least some minimum time (i.e., t_0), which is a strictly positive quantity;
- most of the distribution lies in a small range of time values, depending on σ, which describes the duration of the action under normal conditions;

- the distribution has a tail for $t \rightarrow \infty$ which captures the duration of the action under abnormal conditions.

3. UTILITY OF A PLAN

In this section, we present a method to derive the expected utility of a plan from the value of its goal and the expected duration of the actions. In order to do so, we need first to define the probability that a plan terminates within a given deadline T_{max}.

The probability density function of the duration of a plan can be defined as follows. Given a plan Π_i of a plan library Π and an action α, the probability density function of the plan $\{\Pi_i; \alpha\}$ (i.e., the sequence of Π_i and α) is computed with a convolution of the two *pdf*s associated to Π_i and α:

$$p(t|\Pi_i; \alpha) = p(t|\Pi_i) * p(t|\alpha) \tag{2}$$

In this paper, we consider plans only as sequences of actions, although the formalism can be extended to take more complex structures into account, such as conditional plans with sensing actions. The probability distribution χ^2 is closed under convolution, therefore the distribution describing a plan duration is again a χ^2 distribution. Moreover, its cumulative distribution function (cdf) is a Gamma function which is available in tabular form in many statistical packages.

In general, the desired maximum execution time for a plan T_{max} depends both on the goal achieved by the plan G_{Π_i} and on the current situation in which the robot is, represented by its current *execution state* S_c. Hence, we write this terms as $T_{max}(G_{\Pi_i}, S_c)$ and define the probability of successful termination of plan Π_i from the current execution state S_c as:

$$\Lambda(\Pi_i, t_c, S_c) = P(t_{end}(\Pi_i, t_c) < T_{max}(G_{\Pi_i}, S_c)) \tag{3}$$

In addition to Λ we define a term $U(G_{\Pi_i}, t_c, S_c)$ as the utility of achieving the goal G_{Π_i}, given the robot's execution state S_c at time t_c. It is important to notice that both Λ and U depend on the current time t_c *and* the robot's execution state S_c. Hence, they vary during the plan execution either because the execution state changes or just because time flows.

The expected utility of executing plan Π_i, given S, is defined as

$$U(\Pi_i, t_c, S_c) = U(G_{\Pi_i}, t_c, S_c) \cdot \Lambda(\Pi_i, t_c, S_c) \tag{4}$$

During the execution of the robot task, and consequently the evolution of the current state S_c, the utility values are continuously recomputed since the time variables t_c and T_{max} can change, as well as the parameters of the χ^2 function regulating the actions.

4. ON-LINE EXECUTION MONITORING

The on-line execution monitoring method proposed is based on a probabilistic representation of action duration, and on evaluation of the utility of plans by considering both the utility of the goal achievement and the probability to successfully complete the plan within a desired maximum time.

Algorithm 1 describes in detail the plan execution monitor. The probability distribution of the duration of a plan is constantly evaluated on-line by the robot, based on the current robot execution state, and does not require any re-planning or plan repair procedure. During this step the parameters t_0 and σ of the χ^2 distribution are recomputed according to the execution state of the robot. The computation of the parameters depends on the implementation of the actions, and it is, in general, domain dependent. If at some point the plan with the highest goal utility is unlikely to terminate on time (because the agent is stuck or the environment has changed), its expected utility will drop below the utility of some other plan, and a plan switch will occur. Thus, during execution, we continuously monitor the utility of plans in order to dynamically decide which plan must be executed.

Algorithm 1 PLAN EXECUTION MONITOR

Variables:

Π **:** plan library
$\Pi_{\bar{k}} \in \Pi$ **:** current plan being executed
t_c **:** current time
S_c **:** current state of the robot

procedure $PlanMonitor(\Pi, t_c)$

$\Pi_{\bar{k}} = \emptyset$
while $True$ **do**
  $S_c = getCurrentState(t_c)$
  $\Pi^{\star} = getExecutablePlans(\Pi, S_c)$
  $\Pi_{best} = \mathbf{argmax}_{\Pi_k \in \Pi^{\star} \wedge k \neq \bar{k}} U(\Pi_k, t_c, S_c)$
  if $U(\Pi_{best}, t_c, S_c) - CS(\Pi_{\bar{k}}, \Pi_{best}, S_c) > U^*(\Pi_{\bar{k}}, t_c, S_c)$ **then**
    $\Pi_{\bar{k}} = \Pi_{best}$
    $setCurrentPlan(\Pi_{\bar{k}})$
  end if
end while

5. CONCLUSIONS

We have presented a novel approach to execution monitoring that introduces a probabilistic representation of action duration, and deals with the temporal analysis of actions at execution time. As a consequence, the approach can be applied independently of a specific approach to plan design/generation. The proposed execution monitoring can be embodied in a BDI architecture, where it combines intention selection and plan selection. In this respect, it specifically addresses some of the challenges that have been raised in the dynamic reconsiderations of intentions.

We carried out a preliminary experimental analysis whose results couldn't fit the present paper. As a future work we plan to obtain further evaluation results and to investigate the use of plan structures including hierarchical plans or conditional plans. Finally, we aim at addressing the use of learning techniques to produce execution time estimation of action duration.

6. REFERENCES

[1] M. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.

[2] J. Claßen, Y. Hu, and G. Lakemeyer. A situation-calculus semantics for an expressive fragment of pddl. In *Twenty-Second Conference on Artificial Intelligence (AAAI-07)*. AAAI Press, 2007.

[3] G. D. Giacomo, R. Reiter, and M. Soutchanski. Execution monitoring of high-level robot programs. In *KR*, pages 453–465, 1998.

[4] K. Z. Haigh and M. M. Veloso. Interleaving planning and robot execution for asynchronous user requests. In *Autonomous Robots*, pages 148–155, 1996.

[5] L. Li and M. L. Littman. Lazy approximation for solving continuous finite-horizon mdps. In M. M. Veloso and S. Kambhampati, editors, *AAAI*, pages 1175–1180. AAAI Press / The MIT Press, 2005.

[6] J. Marecki, Z. Topol, and M. Tambe. A fast analytical algorithm for Markov decision process with continuous state spaces. In *Proceedings of the Eight Workshop on Game Theoretic and Decision Theoretic Agents (GTDT) held at the Fifth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 2536–2541, May 2006.