A General Approach to Exploit Model Predictive Control for Guiding Automated Planning Search in Hybrid Domains

Faizan Bhatti ✉, Diane Kitchin, and Mauro Vallati

School of Computing and Engineering, University of Huddersfield, UK
{faizan.bhatti, d.kitchin, m.vallati}@hud.ac.uk

Abstract. Automated planning techniques are increasingly exploited in real-world applications, thanks to their flexibility and robustness. Hybrid domains, those that require to reason both with discrete and continuous aspects, are particularly challenging to handle with existing planning approaches due to their complex dynamics. In this paper we present a general approach that allows to combine the strengths of automated planning and control systems to support reasoning in hybrid domains. In particular, we propose an architecture to integrate Model Predictive Control (MPC) techniques from the field of control systems into an automated planner, to guide the effective exploration of the search space.

Keywords: Automated Planning · Model Predictive Control · Hybrid Reasoning.

1 Introduction

The application of automated planning to real-world domains has always been a matter of great interest for the research community, and is supported by the large set of available planning engines and knowledge engineering approaches. A very common yet challenging feature of real-world applications is the presence of both Boolean and numeric resources and continuous update processes as part of the problem, which makes them hybrid in nature. Planners at the state-of-the-art, rely on the discretisation of dynamic equations of the continuous part of the domain. Due to this discretisation issue, tackling the continuous aspects of hybrid domains is a challenging task for automated planners. On the other hand, planning approaches are extremely performant in dealing with discrete variables, and discontinuities.

In contrast with automated planning, Control Theory is a particularly effective approach for controlling dynamical systems with continuous aspects. Control System techniques are efficient in presence of linear continuous processes or linear-time-invariant (LTI) systems but when there is some non-linearity, it becomes a difficult problem for control system engineers to solve. Given the complementary strengths of control theory techniques and automated planning, it naturally raises the possibility of combining them for better dealing with applications where hybrid reasoning is needed. The idea has been initially explored

by Jimoh [7]. However, the work of Jimoh is domain-specific, and can not be transferred or re-used in different application domains.

In this paper, we propose a general approach to exploit Model Predictive Control (MPC) [2] for guiding automated planning search in hybrid domains. More specifically, we propose an architecture that allows to use MPC techniques from the field of control systems, for calculating a heuristic for the continuous part of a domain model. The heuristic is then exploited by the planner, that has therefore a better overall view of both continuous and discrete elements of the problem at hand. Our experimental analysis, that focuses on a well-known benchmark domain for hybrid PDDL+ planners, demonstrates that our domain-independent approach can improve the planning performance of state-of-the-art planners on this complex type of problems.

2 Background

Automated Planning is a deliberation process which looks at finding a sequence of actions suitable for transforming a given initial state of a system under consideration into a desired goal state. A *planning domain model* is specified by the set of available actions and their consequences whereas a *planning problem* is composed by the domain model, and a description of the initial state, involved objects, and the desired goal. A *solution plan* is a sequence of actions such that their consecutive application, starting from the initial state, results in a state that satisfies the goal [12].

PDDL+ [5], an extension to McDermott's PDDL [8], introduced new modelling features which make it possible to model continuous numeric change in a more realistic way. It introduced the concept of autonomous *processes* and *events*. In PDDL+, a process represents a continuous numeric change with time while maintaining the logical state of the system. A process is autonomous in nature and is not under the direct control of the executive.Similar to processes, events are not under the direct control of the executive. They can be a consequence of a change in the state of the world. Whenever its preconditions are satisfied, an event must occur. The concept of autonomous continuous process made it possible to model many real life problems which were not realisable before. UPMurphi [4], Discretised Nonlinear Heuristic Planner (DiNo) [9], COLIN [3], SMTPlan+ [1] and Expressive Numeric Heuristic Search Planner (ENHSP) [11] are some of the significant planners developed after the introduction of PDDL+.

Model Predictive Control does not refer to a unique control algorithm, but indicates a family of approaches sharing the same philosophy. It starts with the dynamical model of the system and the initial state. At every time step, it calculates the suitable control actions, by solving an open-loop optimisation problem for a given prediction horizon. The prediction output is a sequence of next states and an array of suitable inputs for the system. Only the first control input is picked and applied, so the system can reach the first predicted state. The difference between the predicted state and the actual state is fed back to the controller for correction. This whole process is repeated at every new state.

3 Using MPC to Guide Automated Planning Search

The proposed approach relies on the idea of using MPC for guiding the planning search. In a nutshell, MPC provides a sort of heuristic that can give useful information to the search of a solution, on a given hybrid problem. Fig. 1 depicts the working principle of our approach. The dynamic equations which are related and collectively form a composite entity, which we call a *plant*, are identified and separated. A plant, in this context, is a relatively independent sub-system of the one bigger system. Once a plant is created, the prediction matrices are calculated on the basis of the dynamic model of the plant, which in turns help to calculate the future control moves on the basis of the defined control law.

A general discrete model of a linear-time-invariant system in absence of any input disturbance is given by the following equations.

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

$$y_k = Cx_k + d_k, \tag{2}$$

where x_k is the state vector of the plant, u_k is the control input, y_k is the output, A is the state co-efficient matrix, B is the input co-efficient matrix, C is the output co-efficient matrix, and d_k is the output disturbance at time step k. The discrete model of an LTI system is inherently a one-step-ahead prediction model [10]. Therefore, the output of the system at instance $k+1$ can be determined as

$$y_{k+1} = CAx_k + CBu_k + d_k. \tag{3}$$

The best guess for disturbance is $d_{k+1} = d_k$. Eqs.(1) & (3) represent one-step-ahead relations of state and output vectors. These equations can be used recursively to find any number of step-ahead prediction. In order to calculate the optimum value of future input and in turn the optimum plan, we need to define a control law. We define our performance index or the cost function as a simple quadratic function of error (distance of current state from goal) and the control effort.

$$J = \sum_{k=1}^{N_p} (e_k^T e_k + u_k^T R u_k). \tag{4}$$

Here R is the weighting matrix and defined as positive definite. The error e_k is the difference between the reference value r_k and the actual output y_k. Minimisation of the above performance index with respect to u_k gives us an unconstrained control law for MPC. The addition of constraints from the PDDL+ model makes it a standard quadratic optimisation problem subject to linear constraints and any off-the-shelf quadratic optimiser can be used to solve this quadratic.

To apply MPC in planning we assume that any state of the system is a pair $S = \langle P, N \rangle$, where P is a set of atomic propositions and N is a sequence of numeric values assigned to numeric variables. The input to the planning system is the initial state $I = \langle I_P, I_N \rangle$, the goal condition $G = \langle G_P, G_N \rangle$, the domain model DM and the prediction horizon N_p. Where G_P is a set of atomic

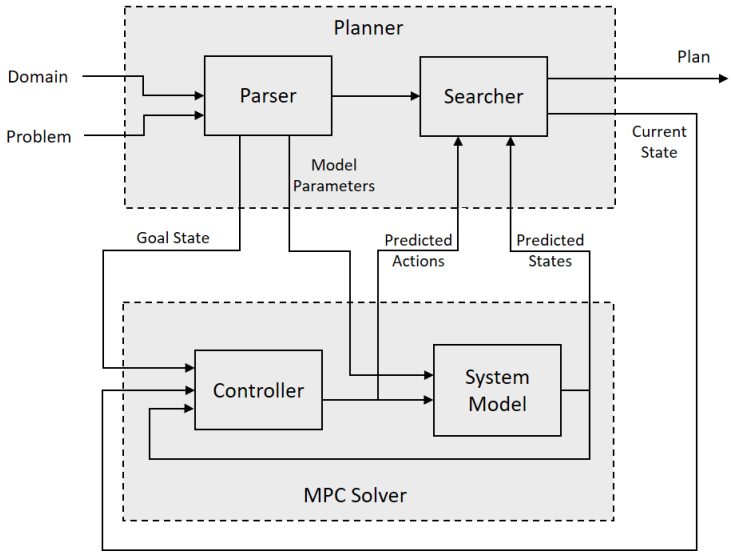

Fig. 1. The proposed architecture to exploit MPC in automated planning.

propositions and G_N is a set of conditions on numeric variables. A node in the search space consists of state S and a path from initial state to the current node which is a partial plan. The search for the solution starts with expanding the nodes from the initial state in classical manner until the preconditions for the processes of the plant are satisfied. As soon as the preconditions for the process are satisfied, MPC solver starts running at that node, considering it as its initial state. The output of MPC solver is a sequence of future actions to be selected and next predicted states as the outcome of those actions. Therefore, it acts as a guide for search. The next node for expansion is selected by finding the node which is nearest to the proposed trajectory of the MPC. If more than one nodes are at the same distance, then any node is selected randomly, since MPC has the tendency to correct itself. Once the next node for the expansion has been chosen, the frontier of search is extended by expanding the chosen node. At this stage, the MPC once again predicts for a new improved trajectory. This process keeps repeating until a solution to G_N is found or the preconditions for processes no longer hold. If G_N is found, the other part of the goal, G_p is searched in classical way. The selection of one node appears to create incompleteness but it limits the search space by pruning a lot of branches and as a result, reduces the search effort.

4 Experimental Analysis

The objective of this experimental analysis is to evaluate the feasibility of using MPC in planning as a heuristic for the continuous processes and to check

Table 1. Performance achieved by ENHSP, and the same planner exploiting the proposed approach (MPC) on instances of the Car domain.

Disp.	Vel. limit	Acc. limit	Planner	Duration	Plan length	Nodes exp	Time(mSec)
30	-10	-2.0	ENHSP	11.0	19	32	230
	+10	+2.0	MPC	12.0	24	28	270
30	-10	-4.0	ENHSP	11.0	17	32	230
	+10	+4.0	MPC	7.0	25	29	250
30	-10	-8.0	ENHSP	11.0	17	32	235
	+10	+8.0	MPC	6.0	36	40	245
30	-10	-10.0	ENHSP	11.0	17	32	235
	+10	+10.0	MPC	6.0	40	44	245
100	-10	-10.0	ENHSP	17.0	27	47	245
	+10	+10.0	MPC	13.0	47	51	265
200	-10	-10.0	ENHSP	27.0	39	519	315
	+10	+10.0	MPC	23.0	57	61	300
500	-10	-10.0	ENHSP	56.0	68	17516	1720
	+10	+10.0	MPC	53.0	87	91	430
1000	-20	-10.0	ENHSP	61.0	71	28512	2440
	+20	+10.0	MPC	56.0	100	104	445

the generality of our methodology. We focus on the well-known Car domain, that is traditionally used for benchmarking the performance of PDDL+ planners, including DiNo [9] and ENHSP [11]. The experiments were carried out in Java NetBeans 8.2 on a machine with Intel(R) Core(TM) i7-8700 CPU, 3.2GHz processor and 16GB RAM. We used IBM CPLEX [6] optimiser for solving the quadratic during the MPC prediction process. While experimenting with different planners, we found ENHSP to be the most efficient planner on these benchmarks. Therefore, our results in this paper are presented in comparison with ENHSP only.

Tab. 1 shows the comparison of ENHSP with MPC in terms of the number of nodes expanded, the duration of the plan, the number of actions (plan length), and the CPU-time needed to find a solution. In the first 4 instances we kept goal distance and velocity constant, while modifying the acceleration limit. Results on those instances shows that the use of our framework does not significantly increase runtime on easy instances, and allows to find solutions of comparable quality. Changing the goal requirement and relaxing the velocity limits shows that the use of MPC allows to outperform ENHSP in terms of expanded nodes, planning time and quality of the generated plans. This is due to the fact that MPC is an optimal control method, so on complex instances it can help in identifying better solutions. An important point to note in results is that the number of nodes expended by MPC approach are always plan length plus 4. As this car domain is a hybrid domain, these 4 nodes refer to the searching for discrete part of the domain, whereas for the continuous part, no node has been expanded which is not part of the final solution plan. This suggests that our

approach can dramatically reduce the search space exploration, and is reflected in the reduced runtime on larger and more complex instances.

5 Conclusion

In this paper, we introduced an approach for exploiting MPC to guide AI planning forward chaining search in general PDDL+ hybrid domains. We demonstrated the effectiveness of the proposed approach on a well-known benchmark domain, that involves coupled equations. Despite a limitation in the implementation that no heuristic has been used for the discrete part of hybrid domain, experimental results showed that our approach can outperform a PDDL+ state-of-the-art planner. The simplicity of our approach is that it can be augmented with any other state of art heuristic search method for classical planning to create a complete hybrid planner. Future works will be focused in considering more benchmarks to better evaluate our approach and in adding a state of the art heuristic to further improve the performance.

References

1. Cashmore, M., Fox, M., Long, D., Magazzeni, D.: A compilation of the full PDDL+ language into SMT. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
2. Clarke, D.W., Mohtadi, C., Tuffs, P.: Generalized predictive control Part I. The basic algorithm. Automatica **23**(2), 137–148 (1987)
3. Coles, A.J., Coles, A.I., Fox, M., Long, D.: Colin: Planning with continuous linear numeric change. Journal of Artificial Intelligence Research **44**, 1–96 (2012)
4. Della Penna, G., Magazzeni, D., Mercorio, F., Intrigila, B.: UPMurphi: a tool for universal planning on PDDL+ problems. In: Nineteenth International Conference on Automated Planning and Scheduling (2009)
5. Fox, M., Long, D.: Modelling mixed discrete-continuous domains for planning. Journal of Artificial Intelligence Research **27**, 235–297 (2006)
6. ILOG, I.: CPLEX optimizer. En ligne]. Available: http://www-01. ibm. com/software/commerce/optimization/cplex-optimizer (2012)
7. Jimoh, F.: A synthesis of automated planning and model predictive control techniques and its use in solving urban traffic control problem. Ph.D. thesis, University of Huddersfield (2015)
8. McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., Wilkins, D.: PDDL-The planning domain definition language (1998)
9. Piotrowski, W.M., Fox, M., Long, D., Magazzeni, D., Mercorio, F.: Heuristic planning for hybrid systems. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
10. Rossiter, J.A.: Model-based predictive control: A practical approach. CRC press (2003)
11. Scala, E., Haslum, P., Thiébaux, S., Ramirez, M.: Interval-based relaxation for general numeric planning. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence. pp. 655–663. IOS Press (2016)
12. Vallati, M., Chrpa, L., Kitchin, D.: ASAP: An automatic algorithm selection approach for planning. International Journal on Artificial Intelligence Tools **23**(06), 1460032 (2014)