Adapting Control Policies to User Preferences

Kristof Van Moffaert[1], Yann-Michaël De Hauwere[1], Peter Vrancx[1], and Ann Nowé[1]

***Abstract*— When designing controllers for machines that interact with human users, it often becomes necessary to adapt control policies to user preferences, even when these preferences are not aligned with the optimal policy. In this paper, we present a reinforcement learning approach that allows to take into account both a classical control performance and end-user feedback. We aim to learn policies that adapt automatically to the needs of a set of users, that rely on the devices. These human-friendly schedules accommodate to user-specific requirements, while simultaneously minimizing operational costs.**

I. INTRODUCTION

Tailoring devices to the needs of a (group of) user(s) has received a great deal of attention in recent years. In settings where devices interact with humans, control objectives often have to be adapted to end-user preferences. We present an approach that views this setting as a multi-objective learning problem. Each objective is represented by a cost function that needs to be minimized. The first objective is a traditional cost function, which specifies the control targets. For example, this cost function can take into account deviation from a target set-point or device energy consumption, depending on the control problem at hand. The second cost function represents the user preferences and is inferred from interactions between users and the device. We assume that human users have the ability to override the current control action and can adapt the policy to their needs. Whenever a human user overrides the suggested action, a penalty is received. From these penalties we deduce the second cost function, which details human preferences.

The final policy should now incorporate both the needs of the users and the cost function. The goal is to find a compromise solution that balances both objectives, without requiring the users to provide information on their preferences beforehand.

II. EXPERIMENTAL RESULTS

As a simple demonstration we use an office espresso machine. The control objective for this machine is to minimize energy consumption by turning the device off when it is idle. However, if the machine is turned off when a user requests coffee, a warm-up period is required, reducing user satisfaction. A basic policy which anticipates every request, consists of an *always-on* profile. Such a naive schedule would obviously have costly consequences both in terms of economical expenses and wear and tear of the machine.

Our multi-objective setting consists of two reward signals, one to indicate the user convenience level and one to indicate the amount of energy being consumed. The former signal depends on the user satisfaction regarding the policy, i.e. positive if the device is ready when a consumption is requested, and negative if the device is turned off at such times. The latter signal represents the economic cost of the schedule. Using appliance monitoring devices, this cost signal was measured accurately. Both reward signals are combined by a weighted-sum and by specifying emphasis on each of the objectives, one can obtain schedules for different trade-off situations, e.g. does the focus lie more on satisfying the user or keeping the economical cost down.

Over a period of one month, we obtained time-based data on the user presences and consumptions [1]. This data,was then offered to the Fitted Q-iteration (FQI) algorithm [2]. FQI is a model-free, batch-mode reinforcement learning algorithm and is particularly suited for problems with large input spaces and large amounts of data.Through an iterated learning process, this algorithm yields a schedule where it has to decide on when to power on and for how long.

By placing more and less emphasis on the objective that focuses on the convenience level of the user, we have obtained a user-oriented and an energy-oriented profile. Due to the limited page-count, a visual representation was omitted, but we compare both schedules and a naive always-on profile in the table below. We see that the potential gains in economical cost of both generated schedules are significant compared to the naive always-on schedule. During the learning process, we noticed that the number of human overrides was decreasing as the algorithm's hypothesis on what are good actions became more clear. In the end, the number of human interventions needed was minimised for both generated schedules. The number of overrides for the energy-oriented schedule is quite low as the most busy timeslots are covered.

	Always-on	User-oriented	Energy-oriented
Hours per day	24h	8h	2h50
Cost per year (€)	578.56	192.86	68.25
Manual overrides	0	1.2	2.1

*This research is supported by the IWT-SBO project PERPETUAL (grant nr. 110041).

[1]K. Van Moffaert, Y.-M. De Hauwere, P. Vrancx and A. Nowé are with the Faculty of Computer Science, Vrije Universiteit Brussel, 1050 Brussels, Belgium kvmoffae@vub.ac.be

REFERENCES

[1] K. Van Moffaert, Y.-M. De Hauwere, P. Vrancx and A. Nowé, Reinforcement Learning for Energy-Reducing Start-Up Schemes (2012), in Proceedings of the 24rd Benelux Conference on Artificial Intelligence

[2] Damien Ernst, Pierre Geurts and Louis Wehenkel (2005), Tree-Based Batch Mode Reinforcement Learning, in Journal of Machine Learning Research, Vol. 6, 503–556