Event-Triggered Interactive Gradient Descent for Real-Time Multi-Objective Optimization

Pio Ong and Jorge Cortés

***Abstract*— This paper proposes an event-triggered interactive gradient descent method for solving multi-objective optimization problems. We consider scenarios where a human decision maker works with a robot in a supervisory manner in order to find the best Pareto solution to an optimization problem. The human has a time-invariant function that represents the value she gives to the different outcomes. However, this function is implicit, meaning that the human does not know it in closed form, but can respond to queries about it. We provide event-triggered designs that allow the robot to efficiently query the human about her preferences at discrete instants of time. For both the cases when the human can answer instantaneously and with some bounded delay, we establish the existence of a minimum interexecution time and the global asymptotic convergence of the resulting executions to the solution of the multi-objective optimization problem.**

I. Introduction

Increasing numbers of robots and machines are constantly being developed to become extensions to human capabilities, to accomplish tasks that humans cannot and to help with menial tasks. With the possibilities of the human having to handle multiple robots, autonomy is receiving a lot of attention lately. In many cases, the final goal of autonomy is an absolute one where human involvement with its operation is no longer necessary. However, in some cases, this may not be possible either because of ethical issues or technological ones. For example, surgical robots are not yet advanced enough nor are trusted to operate on humans without supervision. As a result, much research has been devoted to improving the level of autonomy through studying human-robot interactions where humans interact with autonomous agents only in a supervisory manner rather than in a constantly attentive one.

We are particularly interested in robots or groups of robots that are able to accomplish more than one objective for humans. For these robots, it is often possible for one objective to compromise another. Then, the robots face the multi-objective optimization task of having to decide which objective to prioritize in optimizing its action. For instance, a search-and-rescue group of robots will have to address the question of whether to spend resources searching for more victims or to provide help to the located ones. With current technologies, robots do not yet have the capability to come up with a decision on their own. Most autonomous agents are equipped with predefined metrics and criteria set by humans. The disadvantage with using predefined criteria is that it is less adaptive to the situation presented and the resulting action may not be the most preferred. On the other hand, interactive multi-objective optimization methods will be more robust in finding a preferred result. However, if interactive method were used for multi-objective optimization, the level of full autonomy will be lost because it would require a decision maker (DM) to provide preference information. To the best of our knowledge, there is no interactive multi-objective optimization method that address the level of human involvement with the optimization and consider time to be a critical factor. Motivated by these observations, our main goal is to design an interactive multi-objective optimization method using an event-triggered strategy that prescribes when a DM should interact with an autonomous agent so that the operation converge to what she truly desires.

Literature review. We rely on two bodies of work: multi-objective optimization and event-triggered control. Regarding the first, we are specifically interested in interactive techniques [1], which can be grouped into three main categories: the trade-off approach, the reference points approach and the classification method. Algorithmic solutions often combine elements of several of these categories. In the scenarios considered here, the act of optimization itself is tied in with the physical state of the robot and hence global information is not available a priori. As a result, the reference points method and the classification approaches are not directly applicable. In the trade-off approach, most works focus on finding information related to the gradient of an implicit preference function at each iteration. Examples include the GRIST method developed in [2], which works on finding a subjective gradient to project onto the Pareto front as a search direction, and the SPOT method developed in [3], which suggests finding the gradient of proxy functions. However, as mentioned earlier, overlapping between different approaches is common, and most methods require some global information such as the knowledge of the optimizer of each objective function or the knowledge of the Pareto front. In general, few works in the multi-objective literature examine optimal levels of human involvement or design methods where human take a supervisory role in executing the optimization. The closest work along this line is [4], where the DM waits to choose options generated by a machine, and the convergence to the final solutions can be expected in a number of iterations. However, the method can only be applied to linear programming. In the later version

Pio Ong and Jorge Cortés are with Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA, {piong,cortes}@eng.ucsd.edu

of the work in [5] covering convex cases, the convergence guarantees are lost. The other body of work on which we rely on is the literature on opportunistic and event-triggered control, see e.g. [6]–[9] and references therein. The basic idea is to trade computation and decision-making at the agent level for other processes, be it communication, sensing, motion, or actuation, with the goal of more efficiently employ the available resources. These ideas have also found application in solving distributed optimization problems in an efficient ways in the context of networked systems, cf. [10], [11].

Statement of Contributions. This paper formulates a human-robot interactive multi-objective optimization problem. We consider a scenario where a human works alongside a robot to find the optimizer by responding to its queries about the quality of the outcomes. The human cannot express in closed form its value function, but can provide its gradient (this is a convenient abstraction of the ability of the human to express preferences about an outcome being better than another). we consider two cases of increasing complexity: the case when the human can respond instantaneously and the case when there is a maximum delay in providing this information. The contributions of the paper are threefold. The first contribution is the design of event-triggered laws that allow the robot to query the human efficiently given the overarching multi-objective optimization problem. Our design does not require the DM to be constantly attentive to the multi-objective optimization problem and is based on examining the evolution of the cost function along the various objectives. Our second contribution is the establishment of a uniform lower bound on the interexecution time between any consecutive updates, thus ruling out the possibility of an infinite number of updates in a finite amount of time (i.e., Zeno behavior). The explicit dependence of this bound on the various problem parameters informs the applicability of our design in specific real scenarios. Finally, our third contribution is the analytic characterization of the global asymptotic convergence of the human-robot dynamics to the solution of the multi-objective optimization problem. Under strong convexity, we also show that convergence is exponential. We establish the lack of Zeno behavior and the global convergence for both the cases when the human responds instantaneously and with some maximum delay. Simulations illustrate our results. For reasons of space, all proofs are omitted and will appear elsewhere.

Notation. We denote by $\mathbb{N}$ and $\mathbb{R}$ the set of natural and real numbers, respectively. For $n \in \mathbb{N}$, we use the notation $[n]$ to denote the set $\{1, \dots, n\}$. Given $x \in \mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. We denote by $I_n \in \mathbb{R}^{n\times n}$ the identity matrix and by $\|A\|$ the spectral norm of $A \in \mathbb{R}^{m\times n}$. For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, we let $f_i : \mathbb{R}^n \to \mathbb{R}$ denote its ith-component. The function f is locally Lipschitz if, for every compact set $\mathcal{S}_0 \subset \mathbb{R}^n$, there exists a positive constant L, termed Lipschitz constant, such that $\|f(x) - f(y)\| \leq L\|x - y\|$, for all $x, y \in \mathcal{S}_0$. For $f : \mathbb{R}^n \to \mathbb{R}^m$ continuously differentiable, we denote by $J_f : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ its Jacobian matrix. For a twice continuously differentiable, scalar-valued function $g : \mathbb{R}^n \to \mathbb{R}$, we let $\nabla g : \mathbb{R}^n \to \mathbb{R}^n$ and $\nabla^2 g : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ denote its gradient and Hessian functions. The function g is convex if $\nabla^2 g \succeq 0$, strictly convex if $\nabla^2 g \succ 0$, and strongly convex if there exists $\gamma > 0$ such that $\nabla^2 g \succeq \gamma I_n$. Lastly, we denote the composition of functions $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}$ by $g \circ f : \mathbb{R}^n \to \mathbb{R}$, i.e., $(g \circ f)(x) = g(f(x))$ for $x \in \mathbb{R}^n$.

II. Problem statement

This section describes the problem we set out to solve. We consider a human-robot system that seeks to solve a multi-objective optimization problem formulated as follows. For a vector-valued, continuously differentiable function, $f : \mathbb{R}^n \to \mathbb{R}^m$, consider the unconstrained optimization

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x). \tag{1}$$

A point $x^* \in \mathbb{R}^n$ is a solution of (1) if there does not exist $x \in \mathbb{R}^n$ with $f_i(x) \leq f_i(x^*)$ for all $i \in [m]$ with at least one inequality being strict. These solutions, called Pareto points, capture the fact that improving the minimization of one component of f cannot be done without increasing the value of another. In principle, there exist multiple Pareto points corresponding to the different trade-offs in optimizing the various components of f.

In our model here, a human operator assists the robot in selecting the most appropriate Pareto point. As is commonly done in tradeoff approaches to multi-objective optimization problems, see e.g., [1], [12], [13], we assume that the human has a scalar-valued, continuously differentiable function $v : \mathbb{R}^m \to \mathbb{R}$ that ranks the different outcomes, i.e., $v(f(x))$ represents the 'value' that the human gives to the outcome $f(x)$ achieved at $x \in \mathbb{R}^n$. This function can then be used to establish a preference among all Pareto points. However, the function v is implicit, meaning that the decision maker (DM) does not know it in closed form, but can respond to queries about it. Specifically, we model the human as being able to express preferences about an outcome being better than another one, and we abstract this with gradient information of v: if the robot queries the human about its current value $f(x)$, the human can provide the value $\nabla v(f(x))$. In our technical treatment below, we consider two cases of increasing complexity: the case when the human can respond instantaneously and the case when there is a maximum delay $D > 0$ in providing this information.

With the above model in place, the optimization problem consists of maximizing $v \circ f$. For convenience, we instead formulate this as a minimization problem by considering the cost function $c : \mathbb{R}^m \to \mathbb{R}$, with $c = -v$. The problem to solve is then

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ (c \circ f)(x). \tag{2}$$

We assume the objective function f and the cost function c are strictly convex and bounded below, and that their composition is radially unbounded. Under these assumptions, there

is a unique minimizer x^* to the optimization problem (2). A natural way to find the solution is then to implement the gradient descent algorithm

$$\dot{x} = -\nabla(c \circ f)(x)^T = -(\nabla c(f(x))J_f(x))^T,$$

which is globally convergent to the minimizer. The robot knows the objective function f and can therefore compute its Jacobian, J_f. However, $\nabla c \circ f$ can only be provided by the DM because only she knows the nature of the cost function. Therefore, the implementation of the dynamics requires the human to continuously relay preference information to the robot, which is not feasible. The discretization of the dynamics with a constant stepsize would make its implementation plausible, albeit it will still require periodic human involvement. Given that the stepsize needs to be sufficiently small to guarantee convergence for arbitrary initial conditions, this may still impose an unnecessary load on the human. To tackle this problem, the basic premise of this paper is to endow the robot with criteria that allow it to determine, in an opportunistic fashion, when to query the human to avoid the unnecessary involvement of the DM.

III. Delay-Free Gradient Update Trigger Design

In this section we synthesize a triggering condition for the robot that allows it to efficiently query the human about her preferences regarding the optimization of the vector-valued objective function. We assume that the human can respond to queries immediately, i.e., there is no delay in obtaining the value of $\nabla c \circ f$. Our starting point is the following gradient dynamics discretizing the human component but maintaining the continuous evolution of the robot component, i.e.,

$$\dot{x} = -(\nabla c(f(x_k))J_f(x))^T, \quad t_k \leq t \leq t_{k+1}, \tag{3}$$

where x_k is a shorthand notation to represent $x(t_k)$. Under this dynamics, the human operator only needs to evaluate the robot performance at the discrete time instants $\{t_k\}_{k=0}^{\infty}$. Our goal is then to design a trigger that the robot can evaluate on its own and use to determine this sequence efficiently while still guaranteeing the asymptotic convergence to the desired solution and the feasibility of the resulting implementation.

Our trigger design is based on analyzing the evolution of the cost function evaluated on the objectives towards its optimal value. We consider then

$$V(x) = c(f(x)) - p^*, \tag{4}$$

where $p^* = c(f(x^*))$ denotes the optimal value. Note that, because of our strict convexity assumption, V is positive definite. The next result identifies a gradient update triggering condition that ensures that V is monotonically decreasing at any point other than the optimizer.

Proposition 3.1: (Gradient Update Triggering Condition): Consider the event-triggered human-robot system (3). For each $k \in \{0\} \cup \mathbb{N}$ and $t \geq t_k$, let $\Delta x_k = x(t) - x_k$ denote the error between the state at time t and the state when the gradient was last updated at time t_k. Given the initial state x_0, let L_c be the Lipschitz constant of $\nabla c \circ f$ on the compact set $x \in \mathcal{S}_0 = \{x \mid V(x) \leq V(x_0)\}$. For $\sigma \in (0,1)$, let t_{k+1} be determined according to

$$t_{k+1} = \min\Big\{t \geq t_k \mid \|\Delta x_k\| = \sigma \frac{\|\nabla c(f(x_k))J_f(x)\|}{L_c \|J_f(x)\|}\Big\}. \tag{5}$$

Then, for all $t \in (t_k, t_{k+1})$, we have

$$\frac{d}{dt}V(x(t)) \leq -\frac{1-\sigma}{(1+\sigma)^2}\|\nabla c(f(x(t)))J_f(x(t))\|^2. \tag{6}$$

Although Proposition 3.1 shows that the function V is strictly monotonically decreasing, one cannot yet conclude from it that the optimizer is globally asymptotically stable. The reason for this is that we have not discarded Zeno behavior, i.e., an infinite number of trigger updates in a finite amount of time. Our next result rules out such possibility by establishing a positive lower bound on the interexecution time.

Proposition 3.2: (Lower Bound on Interexecution Time): For the event-triggered human-robot system (3) with updates determined according to (5) and initial condition x_0, the interexecution time is lower bounded as

$$t_{k+1} - t_k \geq \tau_\sigma = \frac{1}{\beta}\ln\left(1 + \frac{\beta\sigma}{L_c J_{\max}}\right) \tag{7}$$

for all $k \in \{0\} \cup \mathbb{N}$, where $J_{\max} = \max_{x \in \mathcal{S}_0}\{\|J_f(x)\|\}$, $\alpha_{max} = \max_{x \in \mathcal{S}_0, i \in [m]}\{\|\nabla^2 f_i\|\}$, $w_{\max} = \max_{x \in \mathcal{S}_0}\{\|\nabla c \circ f(x)\|\}$, and $\beta = m\alpha_{max}w_{\max} > 0$.

The lower bound on the interexecution time in Proposition 3.2 rules out the possibility of Zeno behavior. From (6) and (7), one can see that larger values of $\sigma \in (0,1)$ correspond to longer interexecution times (and hence, more sparing involvement of the decision maker) at the cost of smaller decrease of the Lyapunov function V (and hence affect the speed of convergence). We would like to note here that the statement we just made is based entirely on its given bounds. Because different values of σ produce different trajectories, it is difficult to give a definitive comparison and prediction on their convergence quality. In any case, combining Propositions 3.1 and 3.2, we deduce asymptotic convergence towards the desired optimizer.

Corollary 3.3: (Global Asymptotic Stability): For the event-triggered human-robot system (3) with updates determined according to (5), the optimizer x^* of (2) is globally asymptotically stable. Moreover, if $c \circ f$ is strongly convex with constant $\gamma > 0$, then

$$V(x(t)) \leq V(x_0)e^{-\frac{2\gamma(1-\sigma)}{(1+\sigma)^2}t}, \tag{8}$$

for all $t \geq 0$.

The implication of Proposition 3.1 is that the design choice of σ affects the magnitude of time derivative of V and therefore, the speed of convergence to the optimizer. At the same time, since we could use the interevent time to schedule the DM's presence, the choice of σ may affect the DM's

workload. σ must be carefully picked while considering these two aspects.

IV. Gradient Update Trigger Design when Subjected to Update Delay

In this section, we build on the developments above to deal with the case when the human does not respond instantaneously to queries from the robot, and instead takes some time in providing this information.

We begin our exposition by formally describing the form that the dynamics takes in this scenario. For each $k \in \{0\} \cup \mathbb{N}$, when the robot asks the human at time t_{k+1} for the evaluation of the gradient ∇c at $f(x_{k+1})$, it takes the human some time, $D_{k+1} \geq 0$ to relay the information $\nabla c \circ f(x_{k+1})$. This means that, up until $t_{k+1} + D_{k+1}$, the robot still uses the information provided in the previous communication with the human, i.e., $\nabla c \circ f(x_k)$. The dynamics with delay are then given by

$$\dot{x} = -(\nabla c(f(x_k))J_f(x))^T, t_k + D_k \leq t \leq t_{k+1} + D_{k+1} \quad (9)$$

The human delays are not necessarily the same across different time instants, but we assume them to be uniformly upper bounded by a known constant $D > 0$, representing the maximum time it takes the DM to relay her gradient information.

Given the model above, it is clear that the robot should not wait until it is absolutely necessary to have the new gradient information available to request it from the human (as it did in the delay-free case), unless it is willing to stop its motion until the human replies back. Instead, the robot should take into account the human delay in responding, and ask in advance. Our way of expressing this mathematically is to introduce a new design parameter, σ' to be used in a trigger such that, when enforced for the above dynamics (9), will achieve the same convergence condition obtained in the trigger design for the non-delay case.

Proposition 4.1 (Gradient Update Trigger): Let D^* be the unique solution to

$$\left(\frac{1+\sigma}{1-\sigma}\right)^2 = \frac{e^{\beta(\tau_\sigma - D^*)} - 1}{e^{\beta D^*} - 1}.$$

Consider the event-triggered human-robot system (9) with maximum delay D satisfying $D < D^*$. For $\sigma \in (0,1)$, let $\sigma' \in (0,1)$ be such that

$$\frac{L_c J_{\max}}{\beta}\left(\frac{1+\sigma}{1-\sigma}\right)^2 (e^{\beta D} - 1) < \sigma' \leq \frac{L_c J_{\max}}{\beta}(e^{\beta(\tau_\sigma - D)} - 1), \quad (10)$$

and define t_{k+1} according to

$$t_{k+1} = \min\left\{t \geq t_k \mid \|\Delta x_k\| = \sigma' \frac{\|\nabla c(f(x_k))J_f(x)\|}{L_c J_{\max}}\right\}. \quad (11)$$

Then, $\|\Delta x_k\| < \sigma\left(\frac{\|\nabla(c(f(x_k)))J_f(x)\|}{L_c\|J_f(x)\|}\right)$ when the new gradient information can be implemented, $t \in (t_{k+1}, t_{k+1} + D)$. In addition, t_{k+1} defined above will occur after the latest possible time of the last implementation of gradient information, i.e., $t_{k+1} > t_k + D$. As a consequence, the bound (6) on the evolution of the Lyapunov function holds for all time, $t \in (t_k + D_k, t_{k+1} + D_{k+1})$.

Similarly to the case without delay, we need to prove this new update trigger given by (11) will not exhibit a Zeno behavior in order to show convergence. We shall once again prove lack of a Zeno behavior by providing a lower bound to the interexecution time.

Proposition 4.2 (Interexecution Time with Update Delay): For the choice of σ' chosen to satisfy Proposition 4.1 with the gradient implementation delay of D, the interexecution time lower bound is given by.

$$t_{k+1} - t_k \geq \tau_{\sigma'} = \frac{1}{\beta}\ln\left(\frac{1 + \beta\frac{\sigma'}{L_c J_{\max}}}{1 + \left(\frac{1+\sigma}{1-\sigma}\right)^2 (e^{\beta D} - 1)}\right) + D \quad (12)$$

Having established the absence of Zeno behavior and the fact that the time derivative of the Lyapunov function is the same as the delay-free case, a similar convergence result as Corollary 3.3 follows. The human model with delay improves the practicality of our event-triggered gradient design in realistic scenarios and does not require the robot to have to stop its operation to wait for human to relay information.

V. Simulation Example

Consider a simple robot coverage problem where a robot is required to find an optimal location to observe two different objects at the same time. The objects are located at $p_1 = [0.25; 0.25]$ and $p_2 = [0.21; 0.35]$. Suppose the robot will observe better when it is closer to the objects, the objective functions can be the square of the distances to each object's location.

$$f_1(x) = \|x - p_1\|^2, \qquad f_2(x) = \|x - p_2\|^2$$

In any case, the robot does not know which object is more important to observe and would require a decision maker for assistance. Suppose the DM prefers the robot to be closer to p_1 while still remain in some distance to p_2, and has an implicit cost function of

$$c(f(x)) = 0.8 f_1(x) e^{0.8 f_1(x)} + 0.2 f_2(x) e^{0.2 f_2(x)}$$

Although, she cannot state this function explicitly, if given the distance to each object (the value of the objective functions), she can tell how she would like to decrease one while increase another. This information can be related to the gradient of the cost function with respect to the each objective. We assume there is a method to manipulate the information the DM can give into a perfect information of the human-related gradient, which is given by

$$w(x_k)^T = \begin{bmatrix} 0.8^2 f_1(x_k) e^{0.8 f_1(x_k)} + 0.8 e^{0.8 f_1(x_k)} \\ 0.2^2 f_2(x_k) e^{0.2 f_2(x_k)} + 0.2 e^{0.2 f_2(x_k)} \end{bmatrix}$$

The robot can operate by following the trajectory of $\dot{x} = -(w(x_k)J_f(x))^T$ where

$$J_f(x) = \begin{bmatrix} 2(x-p_1)^T \\ 2(x-p_2)^T \end{bmatrix}$$

Now, suppose that the robot begins at $x_0 = [0.9; 0.6]$, then we know that it is limited to be within the box $[0.21, 0.9] \times [0.25, 0.6]$, and $L_c = 1.1571$ and $J_{\max} = 2.0765$. We use the exact values for the purpose of this simulation, but in reality, one would have to estimate such values, which can be from maximum value achieved in known operating region. Let us assume the parameter $\sigma = 0.3$ was picked to satisfy some convergence rate criteria. Then, from Proposition 3.2, we can calculate the interexecution time of a non-delay dynamics to be $\tau_\sigma = 0.0892$ units. From this value, we can find from Proposition 4.1 that the gradient implementation delay, D, must be less than $D^* = 0.0227$ units. We assume that the DM will take at most $D = 0.0225$ units to implement, and that her performance varies between $[0.5D, D]$. Using criteria (10), we can pick any σ' such that

$$0.2023 < \sigma' \leq 0.2054$$

We picked $\sigma' = 0.2053$. The resulting trajectory is plotted in Figure 1. Figure 2 shows that while enforcing update trigger rules, the human's implicit cost function decreases over time. The mostly linear relationship implies that the implicit cost decreases at an exponential rate. Towards the end of the plot, it may seem that the convergence gets faster as it drops significantly; however, this is a numerical error introduced when last entry of the simulation is taken to be the optimal value. Also, it should be noted here that the function in the figure is not to be compared to the function in the inequality (8) because that was a very conservative result. Next, Figure 3 shows the interexecution time between each requests for humans to update the gradient. This is also plotted in comparison to the lower bound to illustrate that each interevent time lies above the bound.

Finally, we solve the problem with different values of σ. Figure 4 shows the number of human iterations required to reach the one percent of the original cost, for different values of σ. As predicted by the bounds, a bigger σ results in a lower number of triggers. On the contrary, Figure 5 shows an unexpected result. The plot shows the convergence time to one percent of the original cost. Despite the fact that, according to (8), higher σ has a bound that corresponds to slower convergence, in this simulation, higher σ results in faster convergence. This fact points out to the difference between comparing the system execution from the same state and different values of σ for the next triggered time, and comparing the whole trajectories that arise from different values of σ.

VI. Conclusions

In this paper, we have found an update trigger to ensure convergence towards the optimal cost in the human's implicit cost function that weighs each objective in the optimization. We have demonstrated the use of this finding with an example of robot coverage problem. In future works, we plan on refining the model of our human decision maker. First, we would like to introduce errors into the information relayed by the DM, and examine how this may affect the update trigger and the convergence. Next, we may work on introducing time dependency on the cost function such that the DM may be uncertain in the early stage of optimization but becomes more experienced as she interacts with the information relayed by the robot. Lastly, we are also interested in laying out a guideline of how to choose properly the parameter σ, with a consideration of convergence speed and a desire level of

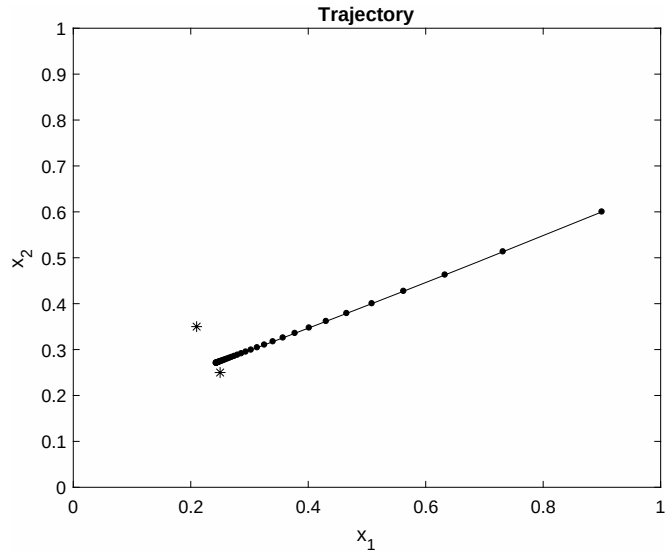

Fig. 1. The robot trajectory with dotted points being where the DM is requested to update the gradient and asterisk points being the two observed objects.

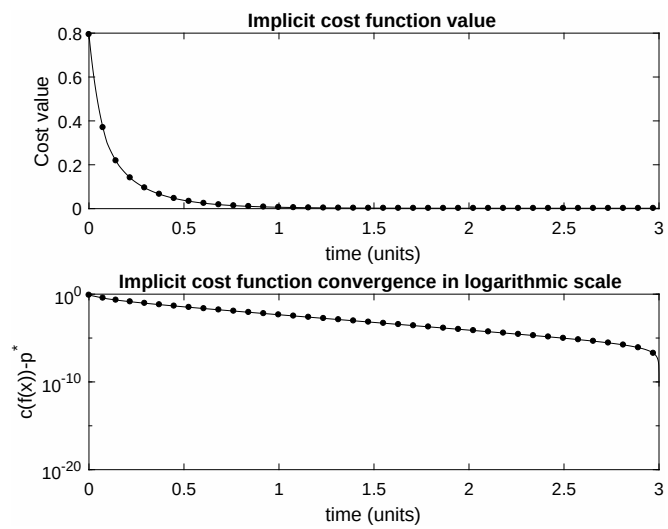

Fig. 2. The plots show the exponential-rate convergence in the human's implicit cost. The markers shows when human update is requested. Also we also show (below) the convergence to the optimal value (taken from last entry in the simulation) in the logarithmic scale. We can see the linear relationship in general between time and the cost.

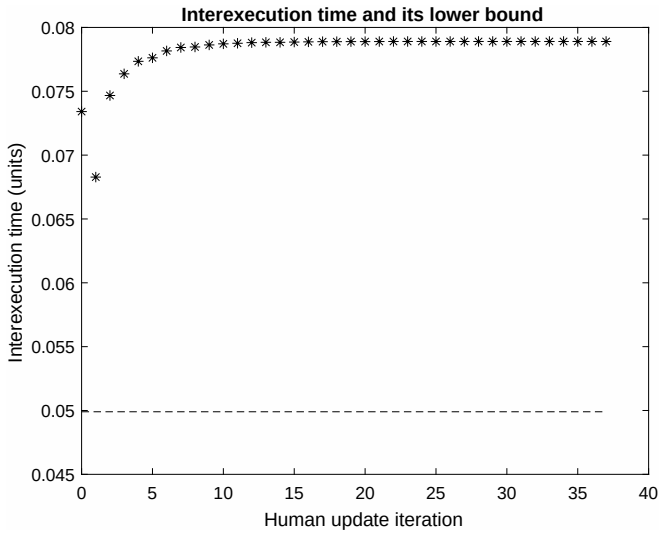

Fig. 3. The interevent time between each request for human to update the gradient value. The lower bound is also plotted in dashed lines.

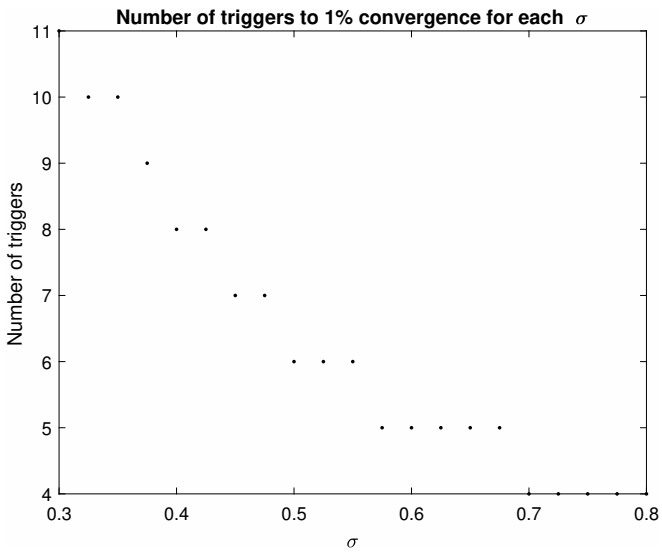

Fig. 4. The number of iterations to reach one percent convergence of the initial Lyapunov's function value for different σ.

DM's workload.

ACKNOWLEDGMENTS

This work was partially supported by NSF award CNS-1329619.

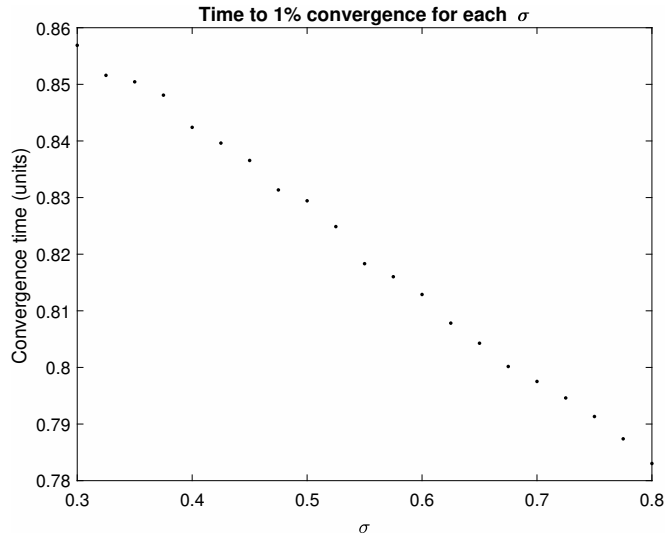

Fig. 5. The amount of time to reach 1 percent convergence of the initial Lyapunov's function value for different σ.

REFERENCES

[1] K. Miettinen, F. Ruiz, and A. P. Wierzbicki, "Introduction to multiobjective optimization: Interactive approaches," in *Multiobjective Optimization: Interactive and Evolutionary Approaches* (J. Branke, K. Deb, K. Miettinen, and R. Słowiński, eds.), pp. 27–57, Berlin, Heidelberg: Springer, 2008.

[2] J. B. Yang, "Gradient projection and local region search for multiobjective optimisation," *European Journal of Operational Research*, vol. 112, no. 2, pp. 432–459, 1999.

[3] M. Sakawa, "Interactive multiobjective decision making by the sequential proxy optimization technique: Spot," *European Journal of Operational Research*, vol. 9, no. 4, pp. 386–396, 1982.

[4] S. Zionts and J. Wallenius, "An interactive programming method for solving the multiple criteria problem," *Management Science*, vol. 22, no. 6, pp. 652–663, 1976.

[5] I. Kaliszewski and S. Zionts, "A generalization of the Zionts-Wallenius multiple criteria decision making algorithm," *Control and Cybernetics*, vol. Vol. 33, no 3, pp. 477–500, 2004.

[6] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 3270–3285, 2012.

[7] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[8] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 586–601, 2011.

[9] M. Miskowicz, *Event-Based Control and Signal Processing*. CRC Press, 2015.

[10] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[11] D. Richert and J. Cortés, "Distributed linear programming with event-triggered communication," *SIAM Journal on Control and Optimization*, vol. 54, no. 3, pp. 1769–1797, 2016.

[12] A. M. Geoffrion, J. S. Dyer, and A. Feinberg, "An interactive approach for multi-criterion optimization, with an application to the operation of an academic department," *Management Science*, vol. 19, no. 4-part-1, pp. 357–368, 1972.

[13] M. Luque, J. B. Yang, and B. Y. H. Wong, "Project method for multiobjective optimization based on gradient projection and reference points," *IEEE Transactions on Systems, Man & Cybernetics. Part A: Systems & Humans*, vol. 39, no. 4, pp. 864–879, 2009.