Autonomy Reconsidered: Toward Developing Multi-Agent Systems

Michael A. Goodrich[1], Julie Adams[2], and Matthias Scheutz[3]

[1] Brigham Young University, Provo, UT 84602, USA
mike@cs.byu.edu
[2] Oregon State University, Corvallis, OR 97331, USA
julie.a.adams@oregonstate.edu
[3] Tufts University, Medford, MA 02155, USA
matthias.scheutz@tufts.edu

Abstract. An agent's autonomy can be viewed as the set of physically and computationally grounded algorithms that can be performed by the agent. This view leads to two useful notions related to autonomy: *behavior potential* and *success potential*, which can be used to measure of how well an agent fulfills its potential, call *fulfillment*. Fulfillment and success potential induce partial and total orderings of possible agent algorithms, leading to algorithm-based, capability-centered definitions of *levels of autonomy* that complement common uses of this phrase. Because the success potential of a multi-agent system can exceed the success potentials of individual agents through synergy effects, the fulfillment of an individual can be augmented through interactions with others, though it can possibly also interfere in the fulfillment of the other agents. *Interaction algorithms* thus enable multiple agents to coordinate, communicate, or exchange information; these algorithms enable and constrain tradeoffs between augmenting and diminishing other agents. Short case studies are presented to illustrate how the algorithm-based definitions can be used to understand existing systems.

1 Introduction

Rapid developments in perception, control, planning, manipulation and navigation enable increasingly advanced robotic systems capable of accomplishing complex tasks autonomously, such as urban driving, traversing rough terrain, or assembling non-trivial products. What does it mean *exactly* for a system to be autonomous and how may that help us to develop increasingly effective and robust systems?

Over the last thirty years many definitions of "autonomy" have explicated what autonomy may mean when applied to artificial systems. Some of these definitions are more detailed and emphasize formally precise conditions, while others provide psychologically and philosophically motivated schemas related to self-governance. One frequently encountered dichotomy is between "automation" and "autonomy", with "automation" roughly referring to fixed action patterns

machines execute without human intervention, regardless of whether the actions achieve the desired effects: "Automation refers to the full or partial replacement of a function previously carried out by a human operator" [38]. A toaster, for example, may ignore the time-dependence of bread type; thus, applying the same duration to all bread types, regardless of easily they may burn.

"Autonomy" is viewed as a system's ability to consider environmental state changes and act upon them (e.g., sensing the correct toasting level, rather than using a fixed time period). For some, an autonomous robot can follow orders, but those orders may leave open exactly what steps are necessary to achieve the task (e.g., [15]). For others, autonomy represents "[an] agent's active use of its capabilities to pursue its goals, without intervention by any other agent in the decision-making processes used to determine how those goals should be pursued" [4]. Other approaches view autonomy on a scale (e.g., "sliding autonomy", "levels of autonomy", "adjustable autonomy"), not as a binary notion. Systems can have degrees of autonomy based on the current context. For example, a clothes dryer with a moisture sensor can adapt the heat levels by sensing dryness, but can be forced by the human to apply a fixed heat level, thus reducing the machine's ability to control the heat adaptively. Similarly, an airplane's auto-pilot attempts to maintain a designated glide path until it no longer can guarantee the path due to, say, bad weather and disengages. Finally, other definitions stress an agent's sensing and actions in an environment and the agent's ability to realize its goals. For example: "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed" [24].

While these approach contain essential elements, particularly the notions of sensing and acting in a dynamic environment in the interest of goals, they lack the precision to capture the important interactions among goals, algorithms, and an agent's physical aspects. Most importantly, notions of task, goals, and success require definition in order to evaluate an agent's performance.

The paper's primary contributions are (a) algorithm-based definitions of behavior potential, success potential, and fulfillment for an individual agent, (b) an argument that interaction between multi-agent systems are potentially more powerful than an autonomous agent, with precise definitions of how interaction algorithms determine synergy, interference, and augmented capability, and (c) short examples that illustrate the utility of the definitions.

2 Related Literature

Beer et al. provide an overview of the notion of autonomy from multiple fields, including philosophy, psychology, and robotics [5]. A common theme is defining a robot's capability in the context of a team's capability, namely a human-robot team. For example, Harbers, Peeters, and Neerincx use an operational definition that implies three specific qualities associated with autonomy: "the time interval of interaction, the obedience of the robot and the informativeness

of the robot" [23]. These qualities include the robot's (1) *capability* and (2) the degree of robot reliance on the *relationship* between the robot and a human partner.

The capability-relationship pair for human-robot and human-agent teams is a pattern in many autonomy definitions. Hexmoor emphasizes the pattern by suggesting that autonomy is "a social notion"; therefore, a robot's autonomy is best defined by the interactions between the robot and some other entity [24]. He writes:

> [A]utonomy concerns are predominantly for the agent to acquire and to adapt to human preferences and guidance ... The word 'autonomous' connotes ... a sense of the agent's autonomy from the human. A device is autonomous when [it] faithfully carries preferences and performs actions accordingly.

Naturally, others have written about independence from and interdependence between agents. Newell writes [37, p. 20]:

> One aspect of autonomy is greater capability to be free of dependencies on the environment ... [but] much that we have learned ... speaks to the dependence of individuals upon the communities in which they are raised and reside.

For example, Dorais says that an autonomous robot can follow orders, but those orders may leave open exactly what steps are necessary to achieve the task [15].

Hexmoor's social notion provides insight into Sheridan's levels of autonomy [38, 47]. Specifically, Sheridan's levels are not explicitly based on a robot's capabilities in the way that Harbers, Peeters, and Neerincx define autonomy. Sheridan's levels implicitly assume a level of capability and explicitly specify properties of the relationship — who has the responsibility for initiating, terminating, or intervening in the behavior induced by the algorithm(s).

Other approaches view autonomy on a scale (e.g., "sliding autonomy", "levels of autonomy", or "adjustable autonomy"), not as a binary notion [47, 29]. There are many autonomy scale variations, and most imply that autonomy is primarily a social notion [7, 14, 16, 18, 21, 28, 27, 31, 35]. Most variants require "social contract" algorithms that enable a human, a robot, or both to (re)assign responsibility/authority for initiating, executing, and terminating functions, information exchanges, and tasks [21, 33].

Dialogues, safeguarding, and shared control are means of designing algorithmic social contracts so that team capability is maximized [17, 18]. For example, shared control seeks to design algorithms that directly support the human-robot team [13, 44, 36]. Naturally, the scope of interaction algorithms can be very large, especially for large multi-agent systems [11, 30, 9, 19]. Social contract algorithms may augment some agents and interfere with others. Shell and Matarić [46] identify one interference type: "Traditional homogeneous foraging has each robot searching for pucks and independently transporting them to the home region ... [A]round the home region; many robots will attempt to enter the same space ... [so] additional robots may hamper the collective effort." Algorithms have been written to mitigate spatio-temporal interference [20]. Sensing interference can also occur [9].

Johnson is critical of contemporary thinking on autonomy [26] and proposes "coactive design," which develops capabilities and algorithms enabling humans and robots to interact well by supporting a form of mutual interdependence.Coactive design includes a specific approach for constructing the social contract so that it explicitly maximizes team capacity. Beer, Fisk, and Rogers propose an approach grounded in function allocation: identify tasks to be performed, determine what task components a robot will perform and is capable of performing, and create a means for a human to influence the robot [5]. Riley proposes a function allocation method that specifies general categories for the types of tasks, information exchanges, and required human-automation interactions [43]. Johnson's, Beer et al.'s, and Riley's approaches directly support systematic design of the social contract algorithm.

Others disavow the social notion of autonomy, emphasizing that autonomy represents "[an] agent's active use of its capabilities to pursue its goals, without intervention by any other agent in the decision-making processes used to determine how those goals should be pursued" [4]. Such definitions emphasize an independence from human input, stressing a robot's sensing and actions in an environment subject to the robot's ability to realize its goals. For example: "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed" [24].

Bradshaw et al. [6] emphasize capability, independent of a social context. They posit two properties essential for an autonomous system: "self-sufficiency, the capability of an entity to take care of itself" and "self-directedness, or freedom from outside control". Similar properties of autonomy appear in non-robotics research (e.g., levels of autonomy for nurse practitioners [10]). Bradshaw's two elements imply that a robot must be able to perform some set of tasks, while also initiating, terminating, and modifying what tasks it performs and how those tasks are performed. Huang et al. [25] similarly state that goals (not a social contract) will govern how capabilities are used: "[A robot's] autonomy [is defined] as its own capability to achieve its mission goals." Beer, Fisk, and Rogers also emphasize self-directedness.

Robot capability, self-sufficiency, and self-directedness must ultimately be implemented as algorithms. Maes [34] presented autonomy as a computational system. Hexmoor's characterization of Maes' work makes the computational system explicit [24]:

> Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously ... realize a set of goals or tasks for which they are designed.

3 Behavior, Success, and Autonomy

A general definition for task environment grounds the discussion. Let $\mathcal{E} = \langle S, I, G, F, \tau \rangle$ be a *(task) environment specification* where S, an environment, is a set of possible states (e.g., a manifold), $I \subset S$ is a set of initial states,

$G \subset S$ is a set of goal states, and F is the evolution function defined on S over τ, where τ is a time bound. Environments are defined as sets of states, to remain as general as possible, while not committing to a particular notion of state or formalism in order to capture many possible environmental states (e.g., a set of differential equations or a Markov decision process) and their relations (e.g., which state is accessible from a given state or whether state transitions are deterministic or stochastic). When needed, the meaning of "state" can be specified (e.g., a six-dimensional kinematic vector, or a set of true propositions at a given point in time) and how exactly they evolve over time (e.g., differential equations, maps, transition functions), including whether the set of time points is discrete or continuous.

The Thermostat as an Example. *Consider an example of maintaining a room's temperature at or near a desired temperature, denoted by θ. There are two relevant states: $S = \{(T < \theta), (T \geq \theta)\}$, where T is the room's temperature. Initial states I can be any room temperature, say $I = [-30, 30]$ C, and goal states are determined by, for example, the goal to "keep the room cool", $G = \{T \leq \theta\}$. An evolution function depends on temperatures outside of the building, the presence of a heating unit, and the presence of an air conditioning (cooling) unit,*

$$F : \begin{cases} T_{t+\Delta t} = T_t + \varepsilon & \text{if heater on} \\ T_{t+\Delta t} = T_t - \varepsilon & \text{if air conditioner on} \\ T_{t+\Delta t} = T_t + \delta(T_{\text{outside}} - T_t) & \text{otherwise.} \end{cases},$$

where Δt denotes a small time step, ε and δ are small positive constants, and T_{outside} denotes the outside air temperature. Finally, τ is some deadline to reach the desired temperature, say $\tau = 20$ min.

Let $\mathcal{R} = \langle P, A \rangle$ be a *robot specification*, where P, the hardware platform includes all sensing, actuating, and computing equipment, and A is an algorithm (plus data) on P that is possibly self-modifying. The sets of sensors *Sen*, effectors *Eff*, and computational systems *Comp* for P are used to define an algorithm as a mapping from sensors/computational states to effector/computational states:

$$A : S_{Sen} \times S_{Comp} \rightarrow S_{Eff} \times S_{Comp}, \tag{1}$$

where the sensor and effector states are the transduced and non-transduced computational interface states, respectively. Computational states, S_{Comp}, include memory, processing, databases, knowledge representations, world models, etc. This formulation permits discussion of the same algorithm on platforms with different sensors, actuators, and representation systems. Computational, sensor, and effector states are part of the environment state,

$$S \supseteq S_{Sen} \cup S_{Comp} \cup S_{Eff}. \tag{2}$$

We differentiate between the instance of the robot's algorithm and the class of algorithms from which the instance is drawn. For example, the class of RRT* algorithms asymptotically approach the optimal solution, but an instance of the RRT* algorithm requires specific parameters (i.e., neighborhood range and cost

function) to generate the robot's behaviors. Similarly, value iteration can find optimal solutions for a Markov-Decision Problem (MDP), but a particular MDP-solver must be instantiated on the robot. The robot's algorithm is an instance of the algorithm class.

There must be a relationship between the robot's algorithm, A, and the evolution function F. If no relationship exists, then the robot has no influence on the environment and autonomy does not matter. The evolution function F includes A as well as other things that influence how world states change: physics, other robots, etc. When discussing autonomy, we are interested in the *trajectories* in the states of environment S *induced by the robot's algorithm* A.

Thermostat Example Continued. *Consider a room that has only an air conditioner and no heater. If the thermostat senses that the current temperature exceeds the desired temperature, it turns on the air conditioner. The thermostat has no memory, so* $S_{Sen} = \{(T < \theta), (T \geq \theta)\}$, $S_{Comp} = \emptyset$, *and* $S_{Eff} = \{on, off\}$. *The thermostat's algorithm is simply*

$$\begin{array}{ll} \textit{if } T \geq \theta \textit{ turn air conditioner on} \\ \textit{else} & \textit{turn air conditioner off.} \end{array}.$$

For temperatures above the desired value, $T_0 > \theta$, *a trajectory is a trace of temperatures falling to the threshold.*

3.1 Absolute Autonomy: Behavior, Success, Fulfillment

A robot's autonomy is determined by the algorithm[4], A, implemented on platform, P. It is conceptually possible to quantify the "amount" of autonomy a robot, $\mathcal{R}$ possesses.

Behavior Potential The *behavior potential* $BP(\mathcal{R})$ of $\mathcal{R}$ in $\mathcal{E}$ is the set of all trajectories in S induced by algorithm A for some starting state $s \in E$ within time bound τ. A "trajectory" is any time-ordered set of states in S determined by how the robot's algorithm A affects the evolution function F, for a given any initial state in S (e.g., flows in dynamical system, state sequences in an MDP). The behavior potential captures all possible behaviors R can exhibit before reaching the time bound τ in any environmental state.

Success Potential Behavior Potential, $BP(\mathcal{R})$, includes two important subsets, $SP(\mathcal{R})$ and $SP^I(\mathcal{R})$. Let $SP(\mathcal{R})$ denote the robot's *success potential*, defined as the the set of trajectories induced by algorithm A, starting from any $s \in E$ leading through a goal state in G within τ. The $SP(\mathcal{R})$ captures all ways for R to succeed at its task. The size of the success potential indicates the robot's capability, and is thus an indicator of potential robot autonomy.

[4] For simplicity of exposition, the set of programs running on a single robot is treated, collectively, as a single algorithm.

Let $SP^I(\mathcal{R})$ denote the robot's *initialized success potential* if the initial state of the environment can be specified, defined as the set of trajectories induced by algorithm A, starting from any $i \in I$ leading through a goal state in G within time bound τ. The difference between $SP(\mathcal{R})$ and $SP^I(\mathcal{R})$ is important because it is easy to create initial environmental states where the algorithm will always fail. For example, start a ground robot in an environmental state where it is dropped from an airplane and the robot will fail.

Thermostat Example continued. *Recall the thermostat algorithm's goal is to cause room temperature to be at or below a desired value starting from any initial value within a time bound τ of 20 minutes. Initial temperatures below the threshold will yield temperatures that are at or below threshold for all time τ (barring some unusual behavior of the evolution function, like a fire in the room). Thus, for states $S_{\text{low}} = \{T \leq \theta\}$ the trajectories are within $SP(\mathcal{R})$. Whether initial states $S_{\text{high}} = \{T > \theta\}$ produce trajectories that are within $SP(\mathcal{R})$ depends on the initial temperature and the laws of thermodynamics. For (a) a time bound τ of sufficient duration, (b) T within the set of feasible states (recall that environment states S included temperatures in the range $[-30, 30]$ C), and (c) an air conditioner of high enough capacity, then (d) all trajectories induced by the thermostat yield success, that is, they are within $SP(\mathcal{R})$.*

The notion of goal state G in $\mathcal{E}$ can be extended when (a) multiple goal states need to be reached by composing multiple tasks or (b) where particular states need to be maintained throughout the task by modeling a subset of S that $\mathcal{R}$ has to maintain. The notion of goal achievement can also be extended for stochastic environments to a probabilistic notion that requires $\mathcal{R}$ end in some goal state, with probability p within τ.

Fulfillment *Fulfillment* is defined as

$$Fulfill = \frac{|SP(\mathcal{R})|}{|BP(\mathcal{R})|},$$

where $|\cdot|$ indicates a set measure, such as cardinality. Fulfillment is a measure of a robot's need to rely on others. Fulfillment measures the proportion of possible initial states for which $\mathcal{R}$ will succeed at its task for a given algorithm A in the absence of help.

Suppose that fulfillment equals one. Then $|SP(\mathcal{R})| = |BP(\mathcal{R})|$, which means that the robot always succeeds – no matter the robot's initial state. A high fulfillment ratio means that the robot does not need to rely on human intervention. The size of the set difference $|BP(\mathcal{R}) \setminus SP(\mathcal{R})|$ is a measure of how often a robot will fail if there is no control over initial conditions; the size of this set measures how much help a robot needs to accomplish its goal.

Thermostat Example continued. *When the goal is simply to keep temperature at or below a threshold, the fulfillment ratio for the thermostat equals one, since success potential equals behavior potential. The thermostat's high fulfillment ratio provides insight into the noted paradox of the thermostat: "A thermostat exercises ... self-sufficiency and self-directedness with respect to the limited tasks*

it is designed to perform through the use of [a] very simple form of automation" [26]. Using this paper's language, the thermostat is autonomous in that it does not need human input (its fulfillment is one), but not in the sense that it is capable of producing many interesting behaviors (its behavior potential is small).

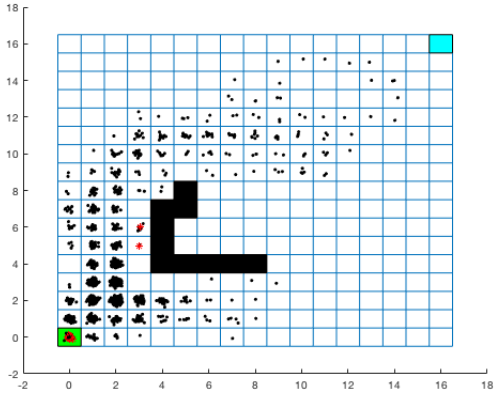

Fig. 1. Low fulfillment.($\rho = 0.4, \gamma = 0.9, \tau = 40$)

A notion similar to fulfillment appears in the literature on reliability and human error in systems[5]. "Operator error probability is defined as the number of errors made ... divided by the number of opportunities for such errors" [40]. Fulfillment emphasizes successful goal-achievement instead of errors.

Fulfillment in a Markov Decision Process Behavior potential, success potential, and fulfillment can be applied to a simple MDP. Consider the grid world shown in Figures 1 and 2. The world states are locations on the grid, $S = \{(x, y) : x, y \in \{0, 1, \ldots, 16\}\}$, the initial state (lower left) is $I = \{(0, 0)\}$, and the goal state (upper right) is $G = \{(16, 16)\}$. The evolution function is a transition probability $p(s'|s, a)$ where s' is the next state, s is the current state, and a is the action specified by the algorithm.

The algorithm is a policy designed to optimize expected discounted reward for some reward structure $R(s, a)$ and some dicount factor γ. The policy maps a sensed state to an action. Thus, the policy implements the definition of an algorithm $A : S_{Sen} \times S_{Comp} \rightarrow S_{Eff} \times S_{Comp}$ as $\pi : S \rightarrow \mathcal{A}$. States are $S_{Sen} = S$, that is, the robot can perfectly sense the world; effectors are $S_{Eff} = \mathcal{A}$, that is, the effector states are the sets of actions that the robot can take; and computation resources,S_{Comp}, is the data structure in which the policy is stored.

[5] Thanks to Karina Roundtree for pointing out the connection between operator error and fulfillment.

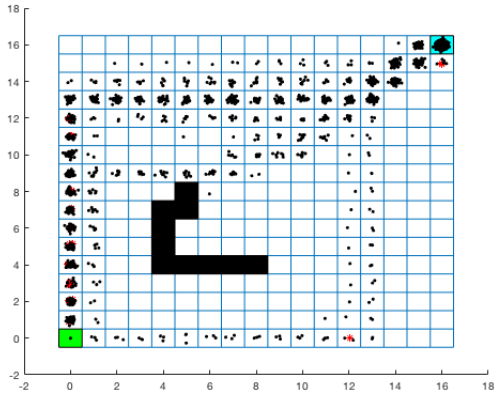

Fig. 2. High fulfillment.($\rho = 0.9, \gamma = 0.999, \tau = 60$)

For concreteness, the following hold: (a) The robot's actions are the cardinal directions, $\mathcal{A} = \{N, S, E, W\}$. (b) The agent moves in the direction it intends (it goes N when $a = N$) with probability parameter ρ and moves in one of the other three directions with probability $\frac{\rho}{3}$. (c) The agent remains in the same position and receives a reward of $r = -1$ when it moves toward a wall. (d) The agent receives a reward of $R = 2$ when it reaches the goal.

Instances of the optimal policy, π, were computed using value iteration. Given a policy, 50 trajectories were computed from the initial condition, generating a sample of the behavior potential. Figure 1 shows the behavior potential for a challenging set of conditions, $\rho = 0.4$, meaning that the the robot goes in an unintended direction $(1-\rho) = 60\%$ of the time. Each dot in a cell represents a visit from the robot in one of the trials. The discount factor was set to $\gamma = 0.9$. The optimal policy for the cells around cell $(2, 2)$ point back to that cell. Essentially, the robot has learned that going through the narrow passageways to the left and below the irregular wall risks a likely collision with a wall, so the "pull" of the goal reward is insufficient to draw the robot through the passageways. For this example, no trajectories reach the goal within $\tau = 40$ time steps, so the success potential is empty. Thus, *Fulfill* $= \frac{0}{50} = 0$.

Figure 2 shows the behavior potential for a policy instance generated from value iteration using the parameters, $\rho = 0.9$, $\gamma = 0.999$, and $\tau = 60$ time steps. The robot moves in the intended direction often, more time is given to complete the task, and the discount factor is high enough to draw the agent to the goal through the narrow passageways. For this world, *Fulfill* $= \frac{50}{50} = 1$.

Figure 3 illustrates fulfillment for various policy instances created from various parameters, yielding three observations. First, when more time is allowed for finding a goal, fulfillment increases. Comparing the left and right figures reveals that, for the same values of ρ and γ, fulfillment is higher when τ is greater.

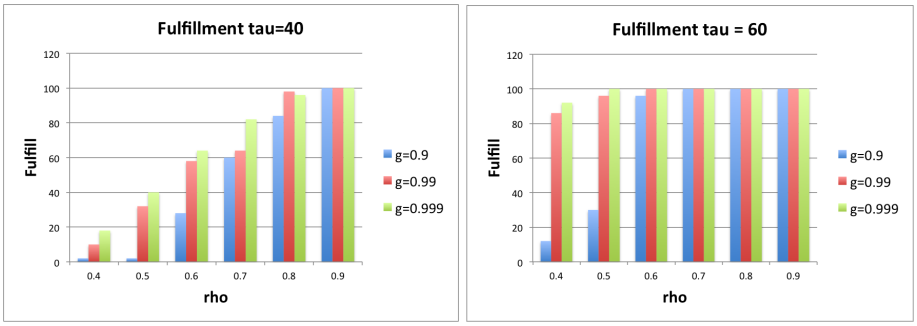

Fig. 3. Fulfillment for $\tau = 40$ and $\tau = 60$, respectively.

Second, even though each policy is optimal for the given discount factor and reward structure, not all algorithm instances have the same fulfillment. In particular, when both γ and ρ are low, the optimal policy tries to stay in the relatively open area in the bottom left, rather than pass through the narrow passageways.

Third, fulfillment depends on both the algorithm through γ and the environment through ρ.

Infinite Sets These examples count the size of finite sets. Infinite sets need a set theoretic measure of set size. Probability measures can be used even if the common perspective that probability represents the frequency of an event precisely is not adopted, because probability measures are special cases of more general set theoretic measures. Future work will demonstrate this claim.

Self-Directedness Some argue that self-directedness is essential for autonomy. The Church-Turing thesis implies that a self-directed agent needs an algorithm or algorithms to select goals, to process knowledge, and to select actions. If self-directedness must be encoded as an algorithm, then success potentials, behavior potentials, and fulfillment apply to that algorithm.

Rationality – An Aside Behavior potential, success potential, and fulfillment are agnostic about whether the algorithm is optimal or rational with respect to some standard. The process by which the algorithm was created is not specified. Because the definitions are agnostic, they complement frameworks that identify optimal algorithms for specific problems. Gerkey and Matarić's taxonomy of independent tasks that can be solved by multi-robot teams [19] is grounded in optimization. The known time and space complexities of algorithms that compute optimal solutions can be used to bound minimum required time budgets τ and what memory resources are required, respectively. Furthermore, knowing the payoff of an optimal solution is useful in trading off the utility of approximate solutions to their fulfillment.

Similarly, the definitions allow for algorithms that are rational with respect to Newell's standard, where he argues that rationality requires an agent pursue a course of action compatible with its goals using knowledge available to the agent [37]. Newell's notion is related to self-directedness, in that a self-directed agent must select goals and pursue those goals using available knowledge.

Being agnostic about how the algorithm is computed may seem to allow irrational agents, and indeed it does. But measuring the fulfillment of irrational agents and comparing against the fulfillment of rational agents allows a comparison of the relative autonomy.

3.2 Relative Autonomy: Levels, Asymmetries, Deficiencies

Given two robots, $\mathcal{R}_1$, and $\mathcal{R}_2$, there are multiple partial or complete orders that can be identified by comparing $SP(\mathcal{R}_1)$ to $SP(\mathcal{R}_2)$ and $SP^I(\mathcal{R}_1)$ to $SP^I(\mathcal{R}_2)$. Intuitively, systems with lower autonomy (in terms of the subset relation) will be able to reach goal states in fewer cases (i.e., from useful initial states) and vice versa.

Recall that capability and non-reliance on others are attributes of autonomy. For the capability attribute, a reasonable definition for *levels of autonomy* (LOA) is:

$$LOA(\mathcal{R}_1) > LOA(\mathcal{R}_2) \textit{ iff } SP(\mathcal{R}_1) \supset SP(\mathcal{R}_2).$$

The LOA is not defined by comparing the fulfillment ratios because fulfillment indicates the potential need for external input or intervention when behaviors cannot be guaranteed to reach the goal. LOAs indicate the relative capability of reaching a goal. We discuss fulfillment in the next section.

MDP Example continued. *Multiple optimal policies were computed for the MDP example. Consider three policies:*

- $\mathcal{R}_1$*'s is computed for* $(\gamma = 0.999, \rho = 0.9)$*,*
- $\mathcal{R}_2$*'s policy is computed for* $(\gamma = 0.9, \rho = 0.7)$*, and*
- $\mathcal{R}_3$*'s policy is computed for* $(\gamma = 0.9, \rho = 0.4)$*.*

We ran 50 trials with those policies in a challenging world ($\rho = 0.4$)*. For each trial, each optimal policy was run using the same seed for the random number generator, with different seeds across trials, which approximates running the algorithms under the same conditions.*

Figure 4 illustrates the results for $\tau = 50$ *time steps. The red* $\times$*'s indicate trials where all algorithms failed to reach the goal. The blue* $\square$*'s indicate the two trials where* $\mathcal{R}_3$ *reached the goal; one success occurred in a trial where both* $\mathcal{R}_1$ *and* $\mathcal{R}_2$ *succeed, and one occurred where both* $\mathcal{R}_1$ *and* $\mathcal{R}_2$ *failed. The green* $\circ$*'s represent trials where both* $\mathcal{R}_1$ *and* $\mathcal{R}_2$ *reached the goal. The cyan *'s represent trials where* $\mathcal{R}_1$ *reached the goal and* $\mathcal{R}_2$ *did not.*

Consider the pair $\mathcal{R}_1$ *and* $\mathcal{R}_2$*. Because* $SP(\mathcal{R}_2)$*, enclosed in the green circle, is a proper subset of* $SP(\mathcal{R}_1)$*,* $\mathcal{R}_1$ *has a higher LOA than* $\mathcal{R}_2$*. Now, consider the pair* $\mathcal{R}_1$ *and* $\mathcal{R}_3$*. What is the relationship between their LOAs? Fulfillment for* $\mathcal{R}_1$ *is much greater than fulfillment for* $\mathcal{R}_3$*, but the success potential for* $\mathcal{R}_1$

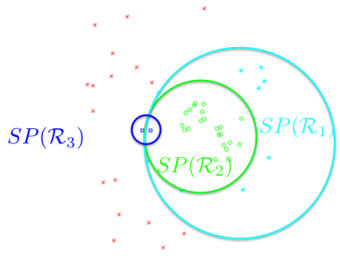

Fig. 4. Success potentials for the MDP problem with different algorithms. The clustering is notional, meaning that it does not represent any environment condition, but is organized to make the sets easy to visualize.

is not a superset of the success potential for $\mathcal{R}_3$. This means that $\mathcal{R}_1$ does not have a higher level of autonomy than $\mathcal{R}_3$, which may seem counter-intuitive. Fortunately, differences in success potentials for different robots can be exploited to maximize group potential.

4 Multi-Agent Systems

Without interference and in the presence of an effective interaction algorithm, the success potential of a group of robots will be at least as high as the union of the success potentials of the individuals,

$$SP(\{\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_n\}) \supseteq \cup_{i=1}^{n} SP(\mathcal{R}_i). \tag{3}$$

Similarly, the behavior potential of a group should be at least has high as the union of individuals, again in the absence of interference,

$$BP(\{\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_n\}) \supseteq \cup_{i=1}^{n} BP(\mathcal{R}_i). \tag{4}$$

4.1 Group Potential: Synergy and Interference

Whether or not the relationships in Equations (3)–(4) hold is subtle for various reasons. First, the number of potential platforms in a team is more than the sum of the individuals. Combined team members can form new platforms (e.g., by connecting [39, 49]) or virtual platforms (e.g., formations [32, 41]). If n robots form the team, then there are 2^n robot combinations of new or virtual platforms.

Second, additional computing resources allow more algorithms. Increased resources increase the number of problems that can be solved (constrained by communications).

Third, entirely new trajectories can be created. For example, trajectories may be enabled that no single robot can perform (e.g., two robots pushing a large box that cannot be pushed by an individual).

Fourth, multiple individual trajectories can be explored simultaneously by a team. For example, ants [22] and honeybees [45] can perform tasks within a time bound that no individual can do by itself within the time bound.

Fifth, the nature of the goal determines which trajectories are successful. Steiner's taxonomy of task types differentiates between unitary tasks and divisible tasks [48]. Divisible tasks can be separated into component subtasks that can each be performed by an individual group member. Unitary tasks must be performed in their entirety, requiring either a coordinated group algorithm or execution by a single team member (or subgroup) with no contributions from others.

Synergy The *synergy potential* from adding more agents $H = \{\mathcal{R}_k : k \in \mathcal{I}\}$ to an existing group of agents $G = \{\mathcal{R}_k : k \in \mathcal{J}\}$, where $G \cap H = \emptyset$, is the set of "extra" things that can be done given the new agents that cannot be done by the original group:

$$\text{Synergy}(H + G) = BP(H \cup G) \setminus \Big(BP(G) \cup BP(H)\Big).$$

This potential can be extended to the extra things that can be done when agents are added to a set of indexed subsets but only the definition for two sets is given for simplicity. Of particular interest is what happens when evaluating what can be accomplished by a group, $G = \{\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_n\}$, when the goal is not divisible,

$$\text{Synergy}^{\text{div}}(G) = BP(G) \setminus \big(\cup_i \mathcal{R}_i BP(\mathcal{R}_i)\big).$$

An analogous definition can be made in terms of success potential, and fulfillment in the presence of synergy can be computed.

Interference Similarly, *interference potential* can be defined as the set of trajectories removed from the subset of trajectories for group G when new agents H are added (e.g., $\mathcal{R}_1$ blocking the path to $\mathcal{R}_2$'s goal location),

$$\text{Interference}(G + H) = BP(G \cup H) \cap BP(G).$$

An analogous definition can be made in terms of success potential, and fulfillment in the presence of interference can be computed.

Synergy and Interference in Swarms Synergy and interference are illustrated using bio-inspired spatial robot swarms. In this example, spatial swarms are composed of simple agents who only interact with their neighbors based on three rules: repulsion, orientation, and attraction. These rules, based on Reynolds's rules for boids [42], are representative of biological swarms [2]. Individual agents' zones of repulsion, orientation, and attraction are centered at an agent's position and are parameterized via the radii of repulsion (r_{rep}), orientation (r_{ori}), and attraction (r_{att}), where $r_{\text{rep}} < r_{\text{ori}} < r_{\text{att}}$. The swarm uses the

topological communication model [3, 1], which assumes an individual can communicate with the N_T nearest agents. Zebrafish have $3-5$ topological neighbors [1], while starlings coordinate with the nearest $6-7$ birds [3]. The examples in this section present results for a topological number of 7 neighbors.

The swarm task is to *search for a goal*, in which the swarm is to locate and move all agents to a single goal location. The goal area's size is scaled to ensure the swarm is able to fit within the goal area. The 600 x 600 pixel world is bounded by a wall that exerts a repulsive force. An agent can sense the goal if it is within the radius of attraction of the goal area's location. Once an agent locates the goal, it communicates the location to its seven neighbors. Agents aware of the goal's location update their headings by equally weighing the desire to travel to the goal and the desire to follow the interaction rules, as was done by others [12, 8].

Synergy and interference are defined using trajectories in the state space. Recall from Eq. 2 that trajectories include computational and effector states. For these spatial swarms, the trajectories include (a) moving from an initial position new locations, (b) forming topological neighborhoods, (c) communicating goal information, and (d) sensing distances and directions of neighbors. Adding agents to a group can create new trajectories in the form of agent networks that communicate information, shaping where agents move.

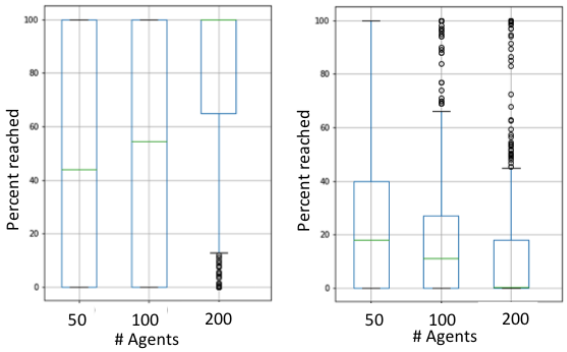

Fig. 5. Fulfillment with swarms for N and no obstacles (left) and 20% obstacles (right).

1,800 simulation trials were conducted, where each trial was 1000 iterations, for 50, 100, and 200 agents. Figure 5 left presents results when there are no environmental obstacles. The percent reached represents the number of agents that reached the goal area, expressed as a percentage of the swarm's size, at the end of the task. This number approximates fulfillment since if 100 robots are in the swarm and 80 reach the goal then 80% of the robots are successful.

In the absence of synergy, if roughly 45% of the 50-agent group reaches the goal then we'd expect roughly the same percent to reach the goal for the 100- and 200-agent groups. Synergy increases fulfillment as agents are added because

more trajectories are possible and successful trajectories are more likely. Larger groups contribute to success in two ways: (a) more agents explore the world, making it more likely that the goal will be found, and (b) more agents tend to form a large connected component through which goal information propagates enabling more agents to reach the goal.

The world becomes more complex by adding obstacles. Obstacle densities of 10% and 20% of the number of agents were evaluated. Both obstacle density levels (10% and 20%) result in a lower percentage of the swarm robots reaching the goal as the number of robots increases. Figure 5 right illustrates decreased fulfillment for 20% density. With 50 robots, an average of 20% of the agents reached the goal, but with 200 robots the average drops to just a small percentage. The precipitous drop in fulfillment is caused by interference. Obstacles "carve" up the large connected component into disconnected connected components, eliminating successful trajectories by preventing goal knowledge to propagate across components.

4.2 Augmentation and Diminishment

When group potential exceeds individual potential, an agent may contribute to the success potential and fulfillment of another agent.

Augmentation *Augmented capability* represents the increase in a agent's success potential when partnered with another agent. Augmented capacity represents what $\mathcal{R}_1$ gains in terms of achieving its goal, when $\mathcal{R}_1$ coordinates with $\mathcal{R}_2$. Augmented capability for $\mathcal{R}_1$ is the increase in goal-achieving trajectories that arises (a) when $\mathcal{R}_2$ induces changes in the evolution function that benefit $\mathcal{R}_1$ (e.g., pushing an obstacle out of the way), (b) when sensor information from $\mathcal{R}_2$ is used as input to $\mathcal{R}_1$'s algorithm (e.g., $\mathcal{R}_2$ provides world state information that $\mathcal{R}_1$ cannot sense), or (c) when $\mathcal{R}_2$'s computational resources are used to solve a problem quickly (within time bound τ) or with a larger amount of memory (e.g., imaging processing). An augmented robot has a higher level of autonomy, because the augmented capacity is defined as an increase in the success potential,

$$SP^{\text{aug}}(\mathcal{R}) \supset SP(\mathcal{R}) \Rightarrow LOA^{\text{aug}}(\mathcal{R}) > LOA^{(}\mathcal{R}),$$

Relying on another can augment an agent's capability.

Diminishment Augmenting a robot's capacity can diminish another's capability. If $\mathcal{R}_1$ requires $\mathcal{R}_2$'s computational resources, then $\mathcal{R}_2$ may be unable to compute what is needed. *Diminished capability* can be defined analogously to augmented capacity, and is a form of interference potential.

Verplank and Sheridan's Levels of Automation Sheridan's LOAs can be revisited in the light of augmentation and diminishment. Figure 6 illustrates a robot's success potential, the green circle surrounding the green ∘'s. The robot

can generate many behaviors, but only a fraction of them generate successes; the cyan *'s are robot failures. If the robot receives human input, such as navigation or perceptual support, then all behaviors will reach the goal, illustrated by the larger cyan circle enclosing the green circle. The success potential grows and fulfillment becomes one. With human input, robot $\mathcal{R}$'s LOA increases because $SP^{\text{aug}}(\mathcal{R}) \subset SP(\mathcal{R})$; the robot, augmented by the human, is strictly more successful.

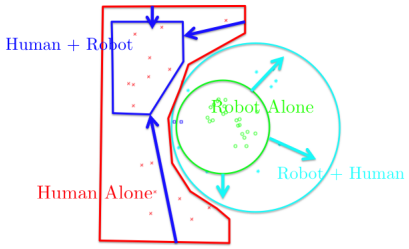

Fig. 6. Augmenting a robot with human input can diminish the human, assuming the human cannot work on other tasks.

Augmenting the robot can cost the human, because human attention and other computational (i.e., cognitive) resources are used to support the robot. Thus, the human splits resources between two tasks, and the resulting set of human behaviors may no longer lead to success. Figure 6 illustrates human success potential without the robot, the red polygon surrounding the red ×'s. When supporting the robot, the human's success potential decreases, the blue polygon. The human's, $\mathcal{H}$, LOA decreases because $SP^{\text{dim}}(\mathcal{H}) \supset SP(\mathcal{H})$.

Whether or not dimishing the human is worth it depends on whether the augmentation benefit is useful. Group potential of non-interacting human and robot is the "size" of the red polygon plus the "size" of the green circle; group potential with interaction is the "size" of the blue polygon plus the "size" of the the cyan circle. Size measures can include the probability of encountering one of the trajectories or the utility of the trajectories or some combination.

Consider four of Sheridan's LOAs: autonomous (Auto, level 10), Do-Then-Tell[6] (Dtt, levels 6–9), Ask-Then-Do (Atd, levels 2–5), and teleoperation (Tele, level 1).

First, assume that Auto means the robot can succeed at all behaviors without human input. Figure 7 left cross-plots $(SP^{\text{dim}}(\mathcal{H}), SP^{\text{aug}}(\mathcal{R}))$ the combined success potentials for various Sheridan-based LOAs. The plots assume it is possible for the autonomous robot to accomplish all its goals from any starting condition – fulfillment is one and success potential is large. What the robot can accomplish sans human help is plotted on the y–axis; the robot wastes time interacting with the human and resorts to autonomous mode. Human-diminishment

[6] Thanks to Lanny Lin for the names of Do-Then-Tell and Ask-Then-Do.

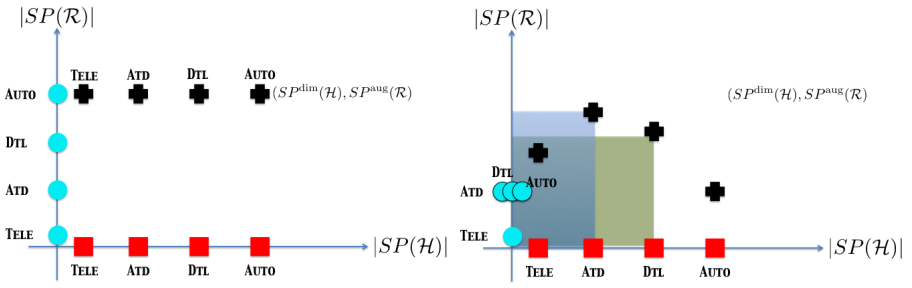

Fig. 7. Success potentials when autonomy is just as capable as a human and robot working together (left) and when autonomy can't achieve what human and robot together can (right). •'s indicate robot success potentials, ■'s indicate human success potentials, and thick +'s indicate group success potentials.

from reduced computation-budget is plotted on the x-axis, with maximum computational resources available if the human is not obligated to assist the robot. The human-robot team elevates all robot LOAs to that of a fully autonomous robot, but at the cost of what the human can do when not assisting the robot. Group success is plotted as black +'s. For this example, AUTO maximizes group fulfillment.

Second, assume that the fully autonomous robot does not achieve maximum fulfillment and lacks sufficient capability to achieve maximum success potential. Figure 7 right cross-plots a diminished human and an augmented robot. Sans human input, DTT and ATD perform the same as AUTO, but they are equipped with a human interaction algorithm that allows them to be augmented. TELE must have human input to perform well. With human help, ATD can achieve maximum fulfillment but with human diminishment. DTT can be augmented with human interaction to achieve high-but-not-maximum fulfillment, with lower human diminishment cost. The shaded rectangles indicate group success potential for ATD and DTT; larger areas indicate larger group fulfillment. For this example, DTT maximizes group fulfillment but ATD maximizes robot fulfillment.

The ideal robot LOA depends on the success potential and fulfillment for the robot, the human, and the group.

5 Summary

This paper provides precise algorithm-based definitions for two attributes of agent autonomy: capability (defined as the size of the success potential set) and nonreliance on another agent (defined using fulfillment). The definitions are extended for multiple agents, leading to notions of synergy and interference. The potential for group capability and fulfillment to be higher than the sum of individuals in the group make it possible to estimate tradeoffs in multi-agent teams; specifically how a contribution from agent A can augment agent B, but at a potential cost in capability and fulfillment for agent A. Case studies were

used to illustrate the definitions, emphasizing how the definitions give insight into existing problems.

Acknowledgment

This work has in part been funded by ONR grant #N00014-18-1-2831.

References

1. N. Abaid and M. Porfiri. Fish in a ring: spatio-temporal pattern formation in one-dimensional animal groups. *Journal of the Royal Society Interface*, 7(51):1441–1453, 2010.
2. I. Aoki. A simulation study on the schooling mechanism in fish. *Bull. Jap. Soc. Sci. Fish.*, 48(8):1081–1088, 1982.
3. M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, 105(4):1232–1237, 2008.
4. S. K. Barber, A. Goel, and C. E. Martin. Dynamic adaptive autonomy in multi-agent systems. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(2):129–147, 2000.
5. J. M. Beer, A. D. Fisk, and W. A. Rogers. Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of Human-Robot Interaction*, 3(2):74–99, 2014.
6. J. M. Bradshaw, P. J. Feltovich, H. Jung, S. Kulkarni, W. Taysom, and A. Uszok. Dimensions of adjustable autonomy and mixed-initiative interaction. In *Agents and Computational Autonomy*, pages 17–39. Springer, 2004.
7. J. M. Bradshaw, M. Sierhuis, A. Acquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok, and R. V. Hoof. Agent autonomy. In H. Hexmoor, R. Falcone, and C. Castelfranchi, editors, *Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications*. Kluwer, 2002.
8. D. S. Brown, M. A. Goodrich, S.-Y. Jung, and S. C. Kerman. Two invariants of human swarm interaction. *Journal of Human-Robot Interaction*, 5(1):1–31, 2016.
9. W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.
10. C. B. Cajulis and J. J. Fitzpatrick. Levels of autonomy of nurse practitioners in acute care setting. *Journal of the American Association of Nurse Practitioners*, 19(10):500–507, 2007.
11. D. Claes, P. Robbel, F. A. Oliehoek, K. Tuyls, D. Hennes, and W. van der Hoek. Effective approximations for multi-robot coordination in spatially distributed tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 881–890. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
12. I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

13. J. W. Crandall and M. A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *Proceedings of the 2002 IEEE /RSJ International Conference on Intelligent Robots and Systems*, Luasanne, Switzerland, 2002.
14. M. B. Dias, B. Kannan, B. Browning, E. G. Jones, B. Argall, M. F. Dias, M. Zinck, M. M. Veloso, and A. J. Stentz. Sliding autonomy for peer-to-peer human-robot teams. In *Proceedings of the Intelligent Conference on Intelligent Autonomous Systems*, 2008.
15. G. Dorais, R. P. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems. In *Working notes of the Sixteenth International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems*, pages 16–35, 1999.
16. M. R. Endsley and D. B. Kaber. Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42(3):462–492, 1999.
17. T. Fong, C. Thorpe, and C. Baur. A safeguarded teleoperation controller. In *IEEE International Conference on Advanced Robotics*, Budapest, Hungary, August 2001.
18. T. Fong, C. Thorpe, and C. Baur. *Collaboration, dialogue, human-robot interaction*, pages 255–266. Springer, 2003.
19. B. P. Gerkey and M. J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
20. D. Goldberg and M. J. Matarić. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings, AAAI-97*, pages 637–642, Providence, Rhode Island, July 1997.
21. M. A. Goodrich, D. R. Olsen, J. W. Crandall, and T. J. Palmer. Experiments in adjustable autonomy. In *Proceedings of the IJCAI01 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, 2001.
22. D. M. Gordon. *Ant encounters: interaction networks and colony behavior*. Princeton University Press, 2010.
23. M. Harbers, M. M. M. Peeters, and M. A. Neerincx. Perceived autonomy of robots: Effects of appearance and context. In *International Conference on Robot Ethics*, Lisbon, Portugal, 2015.
24. H. Hexmoor, C. Castelfranchi, and R. Falcone. A prospectus on agent autonomy. In *Agent Autonomy*, pages 1–10. Springer, 2003.
25. H.-M. Huang, E. Messina, and J. Albus. Toward a generic model for autonomy levels for unmanned systems (ALFUS). Technical report, DTIC Document, 2003.
26. M. J. Johnson. *Coactive Design: Desining Support for Interdependence in Human-Robot Teamwork*. PhD thesis, Technische Universiteit Delft, Delft, The Netherlands, 2015.
27. D. B. Kaber and M. R. Endsley. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science*, 5(2):113–153, March-April 2004.
28. D. B. Kaber, E. Onal, and M. R. Endsley. Design of automation for telerobots and the effect on performance, operator situation awareness and subjective workload. *Human Factors and Ergonomics in Manufacturing*, 10(4):409–430, 2000.
29. D. B. Kaber and J. Riley. Adaptive automation of a dynamic control task based on secondary task workload measurement. *International journal of cognitive ergonomics*, 3(3):169–187, 1999.

30. G. Kaminka, I. Frank, K. Arai, and K. Tanaka-Ishii. Performance competitions as research infrastructure: Large scale comparative studies of multi-agent teams. *Autonomous Agents and Multi-Agent Systems*, 7(1):121–144, 2003.
31. D. Kortenkamp, P. Bonasso, D. Ryan, and D. Schreckenghost. Traded control with autonomous rbots as mixed initiative interaction. In *AAAI Symposium on Mixed Initiative Interaction*, Stanford, CA, USA, 1997.
32. M. A. Lewis and K.-H. Tan. High precision formation control of mobile robots using virtual structures. *Autonomous robots*, 4(4):387–403, 1997.
33. L. Lin and M. A. Goodrich. Sliding autonomy for UAV path-planning: Adding new dimensions to autonomy management. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
34. P. Maes. Modeling adaptive autonomous agents. *Artificial life*, 1(1_2):135–162, 1993.
35. C. A. Miller, H. B. Funk, M. Dorneich, and S. D. Whitlow. A playbook interface for mixed initiative control of multiple unmanned vehicle teams. In *Proceedings of the 21st Digital Avionics Systems Conference*, volume 2, pages 7E4–1 – 7E4–13, November 2002.
36. M. Mulder, D. A. Abbink, and T. Carlson. Journal of human-robot interaction. Special issue on Shared Control, 2015.
37. A. Newell. *Unified theories of cognition*. Harvard University Press, 1994.
38. R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 30(3):286–297, May 2000.
39. R. Pfeifer, M. Lungarella, and F. Iida. Self-organization, embodiment, and biologically inspired robotics. *science*, 318(5853):1088–1093, 2007.
40. R. W. Proctor and T. V. Zandt. *Human factors in simple and complex systems*. CRC press, 2008.
41. W. Ren and R. W. Beard. Decentralized scheme for spacecraft formation flying via the virtual structure approach. *Journal of Guidance, Control, and Dynamics*, 27(1):73–82, 2004.
42. C. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 21:25–34, 1987.
43. V. Riley. FAIT: A systematic methodology for identifying system design issues and tradeoffs. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1989.
44. T. Rofer and A. Lankenau. Ensuring safe obstacle avoidance in a shared-control system. In J. M. Fuertes, editor, *Proc. of the 7th Int. Conf. on Emergent Technologies and Factory Automation*, pages 1405–1414, 1999.
45. T. D. Seeley. *Honeybee democracy*. Princeton University Press, 2010.
46. D. A. Shell and M. J. Mataric. On foraging strategies for large-scale multi-robot systems. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2717–2723. IEEE, 2006.
47. T. B. Sheridan and W. L. Verplank. Human and computer control of undersea teleoperators. Technical report, MIT Man-Machine Systems Laboratory, 1978.
48. I. D. Steiner. Group processes and group productivity. *New York: Academic*, 1972.
49. M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):43–52, 2007.