Acting Deceptively: Providing Robots with the Capacity for Deception

Alan R. Wagner

Georgia Institute of Technology and
Georgia Tech Research Institute
85 5th Street, Room S27, NW,
Atlanta, GA. 30308
1-404-894-9311

alan.wagner@gatech.edu,
alan.wagner@gtri.gatech.edu

Ronald C. Arkin

Georgia Institute of Technology
85 5th Street, S27, NW,
Atlanta, GA. 30308
1-404-894-9311

arkin@cc.gatech.edu

Abstract

Deception is utilized by a variety of intelligent systems ranging from insects to human beings. It has been argued that the use of deception is an indicator of theory of mind [2] and of social intelligence [4]. We use interdependence theory and game theory to explore the phenomena of deception from the perspective of robotics, and to develop an algorithm which allows an artificially intelligent system to determine if deception is warranted in a social situation. Using techniques introduced in [1], we present an algorithm that bases a robot's deceptive action selection on its model of the individual it's attempting to deceive. Simulation and robot experiments using these algorithms which investigate the nature of deception itself are discussed.

Keywords*: deception, game theory, interdependence theory, interaction, hide-and-seek, theory of mind.*

1. Introduction

Deception has a long and deep history with respect to the study of intelligent systems. Biologists and psychologists argue that deception is ubiquitous within the animal kingdom and represents an evolutionary advantage for the deceiver [5]. Primatologists note that the use of deception serves as an important potential indicator of theory of mind [2] and social intelligence [4]. Researchers in these fields point to numerous examples of deception by non-human primates. From a roboticist's perspective, the use of deception and the development of strategies for resisting being deceived are important topics of study especially with respect to the military domain [7].

But what is deception? McCleskey notes that deception is a deliberate action or series of actions brought about for a specific purpose [8]. Whaley recognizes that deception often includes information provided with the intent of manipulating some other individual [9]. Ettinger and Jehiel offer a related definition describing deception as, "the process by which actions are chosen to manipulate beliefs so as to take advantage of the erroneous inferences [10]." This definition has clear ties to game theory but does not relate to many of the passive, unintentional examples of deception found in biology. We adopt a definition for deception offered by Bond and Robinson that encompasses conscious and unconscious, intentional and unintentional acts of deception. These authors describe deception simply as *a false communication that tends to benefit the communicator* [5].

This paper investigates the use of deception by autonomous robots. We focus on the actions, beliefs and communication of the deceiver, not the deceived. Specifically, our central thesis is that modeling of the individual to be deceived is a critical factor in determining the extent to which a deceptive behavior will be effective. In other words, a robot must have specific knowledge about the individual that it is attempting to deceive—*the mark*—in order for the deceptive action to be effective. It is worth noting that a deceiver's knowledge of the mark need not be explicit. The exploration of this thesis is important both for the creation of deceptive robots and for developing a better understanding of the nature of deception itself.

Consider, for example, the use of camouflage. Camouflage is the use of natural or artificial material to allow an otherwise visible object to remain indiscernible from the surrounding environment [7]. The act of camouflaging assumes that the mark has specific perceptual characteristics, such as color vision. A robot that relies on infrared, for example, will not be deceived by color-based camouflaging.

We will use the following example to illustrate our ideas: a valuable robotic asset operates at a military base. The base comes under attack and is in danger of being overrun. If the robot is discovered by the attackers then they will gain valuable information and hardware. The robot must hide and select a deceptive strategy that will reduce the chance that it will be encountered. Throughout this article we will use this running example to explain portions of the theoretical underpinnings of our approach as well as to develop experiments based on the example.

The remainder of this paper begins by first summarizing relevant research. Next, we use game theory and interdependence theory to reason about the theoretical underpinnings of deception and to develop preliminary algorithms for the effective use of deception on a robot. Finally, we present a series of experiments which attempt to investigate the veracity of our thesis. The article concludes with a discussion of these results including directions for future research and the ethical implications of our research.

2. Related Work

Game theory has been extensively used to explore the phenomena of deception. As a branch of applied mathematics, game theory focuses on the formal consideration of strategic interactions, such as the existence of equilibriums and economic applications [11]. Signaling games, for example, explore deception by allowing each individual to send signals relating to their underlying type [12]. Costly versus cost-free signaling has been used to determine the conditions that foster honesty. Floreano et al. found that deceptive communication signals can evolve when conditions conducive to these signals are present [13]. These researchers used both simulation experiments and real robots to explore the conditions necessary for the evolution of communication signals. They found that cooperative communication readily evolves when robot colonies consist of genetically similar individuals. Yet when the robot colonies were genetically dissimilar and evolutionary selection of individuals rather than colonies was performed, the robots evolved deceptive communication signals, which, for example, compelled them to signal that they were near food when they were not. Floreano et al.'s work is interesting because it demonstrates the ties between biology, evolution, and signal communication and does so on a robotic platform.

Ettinger and Jehiel have recently developed a theory for deception based on game theory [10]. Their theory focuses on belief manipulation as a means for deception. In game theory, an individual's *type*, $t_i \in T_i$, reflects specific characteristics of the individual and is privately known by that individual. Game theory then defines a *belief* as, $p_i\left(t_{-i} \middle| t_i\right)$, reflecting individual *i*'s uncertainty about individual -*i*'s type [11]. Ettinger and Jehiel demonstrate the game theoretical importance of modeling the mark. Still, their definition of deception as "the process by which actions are chosen to manipulate beliefs so as to take advantage of the erroneous inferences" is strongly directed towards game theory and their own framework. The question thus remains, what role does modeling of the mark play for more general definitions of deception such as those offered by [5]. As mentioned above, our goal is to explore the phenomena of deception from as general a perspective as possible. Our belief is that by taking this broad approach we will uncover aspects of the phenomena of deception that apply not just to robot-robot interaction, but also to human-robot and interpersonal interaction.

Deception can also be explored from a social psychological perspective. Interdependence theory, a type of social exchange theory, is a psychological theory developed as a means for understanding and analyzing interpersonal situations and interaction [14]. The term interdependence specifies the extent to which one individual of a dyad influences the other. Interdependence theory is based on the claim that people adjust their interactive behavior in response to their perception of a social situation's pattern of rewards and costs. Thus, each choice of interactive behavior by an individual offers the possibility of specific rewards and costs (also known as outcomes) after the interaction. Interdependence theory and game theory represent social situations computationally as an outcome matrix. An outcome matrix represents a social situation by expressing the outcomes afforded to each interacting individual with respect each pair of potential behaviors chosen by the individuals.

3. Representing Interactions

The outcome matrix is a standard computational representation for interaction [14]. It is composed of information about the individuals interacting, including their identity, the interactive actions they are deliberating over, and scalar outcome values representing the reward minus the cost, or the outcomes, for each individual. Thus, an outcome matrix explicitly represents information that is critical to interaction. Typically, the identity of the interacting individuals is listed along the dimensions of the matrix.

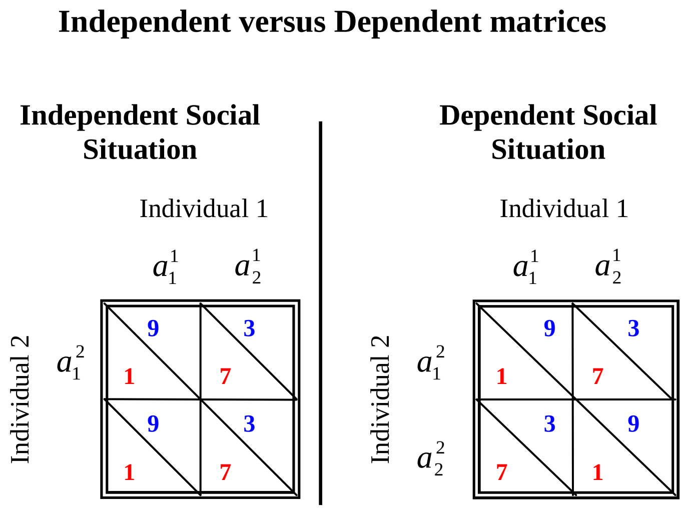


Figure 1. An example of an independent situation is depicted on the left and an example of a dependent situation is depicted on the right. In the example of an independent situation,, the action selection of the second individual does not have an effect the outcome received by the first individual. In the dependent example, on the other hand, the action selection of the second individual results in a gain or lose of 7 units of outcome (a measure of utility) by the first individual.

Figure 1 depicts an interaction involving two individuals. In this article, the term individual is used to indicate a human, a social robot, or an agent. We will focus on interaction involving two individuals: dyadic interaction. An outcome matrix can, however, represent interaction involving more than two individuals. The rows and columns of the matrix consist of a list of actions available to each individual during the interaction. Finally, a scalar outcome is associated with each action pair for each individual. Outcomes represent unitless changes in the robot, agent, or human's utility.

Thus, for example, an outcome of zero reflects the fact that no change in the individual's utility will result from the mutual selection of that action pair.

Because outcome matrices are computational representations, it is possible to describe them formally. Doing so allows for powerful and general descriptions of interaction. The notation presented here draws heavily from game theory [11]. A representation of interaction consists of [15]:

1) a finite set *N* of interacting individuals;
2) for each individual $i \in N$ a nonempty set A^i of actions; and
3) the utility obtained by each individual for each combination of actions that could have been selected.

Let $a_j^i \in A^i$ be an arbitrary action *j* from individual *i*'s set of actions. Let $\left(a_j^1,\ldots,a_k^N\right)$ denote a combination of actions, one for each individual, and let u^i denote individual *i*'s utility function, where $u^i\left(a_j^1,\ldots,a_k^N\right) \to \Re$ is the utility received by individual *i* if the individuals choose the actions $\left(a_j^1,\ldots,a_k^N\right)$. The term O is used to denote an outcome matrix. A particular outcome within a matrix can be expressed as a function of an outcome matrix and an action pair, thus $O^i\left(a_2^i, a_1^{-i}\right) = {}_{12}o^i$. The variable *o* denotes an outcome value. The term ${}_{12}o^i$ denotes that it is individual *i*'s outcome from the first row and second column of the matrix. The superscript *-i* is used to express individual *i*'s partner. Thus, for example, A^i denotes the action set of individual *i* and A^{-i} denotes the action set of individual *i*'s interactive partner. As mentioned above, an individual's *type*, $t^i \in T^i$, is determined prior to interaction, reflects specific characteristics of the individual and is privately known by that individual. A *belief*, $p^i\left(t^{-i}\middle|t^i\right)$, expresses individual *i*'s uncertainty about individual -*i*'s type.

3.1 Representing Social Situations

The term interaction describes a discrete event in which two or more individuals select interactive behaviors as part of a social situation or social environment. Interaction has been defined as influence—verbal, physical, or emotional—by one individual on another [16]. The term situation has several definitions. The most apropos for this work is “a particular set of circumstances existing in a particular place or at a particular time” (World English Dictionary, 2007). A social situation, then, characterizes the environmental factors, outside of the individuals themselves, which influence interactive behavior. A social situation is abstract, describing the general pattern of outcome values in an interaction. An interaction, on the other hand, is concrete with respect to the two or more individuals and the social actions available to each individual. For example, the prisoner’s dilemma describes a particular type of social situation. As such, it can, and has been, instantiated in numerous different particular social environments ranging from bank robberies to the trenches of World War I [17]. Interdependence theorists state that interaction is a function of the individuals interacting and of the social situation [18]. Although a social situation may not afford interaction, all interactions occur within some social situation. Interdependence theory represents social situations involving interpersonal interaction as outcome matrices.

In previous work, we presented a situation analysis algorithm that calculated characteristics of the social situation or interaction (such as interdependence) when presented with an outcome matrix [3]. The interdependence space is a four-dimensional space which maps the location of all interpersonal social situations [19]. A matrix's location in interdependence space provides important information relating to the interaction. The interdependence and correspondence dimensions are of particular importance for recognizing if a situation warrants deception. The interdependence dimension measures the extent to which each individual's outcomes are influenced by the other individual's actions in a situation. In a low interdependence situation, for example, each individual's outcomes are relatively independent of the other individual's choice of interactive behavior (Figure 1 left for example). A high interdependence situation, on the other hand, is a situation in which each individual's outcomes largely depend on the action of the other individual (Figure 1 right for example). Correspondence describes the extent to which the outcomes of one individual in a situation are consistent with the outcomes of the other individual. If outcomes correspond then individuals tend to select interactive behaviors resulting in mutually rewarding outcomes, such as teammates in a game. If outcomes conflict then individuals tend to select interactive behaviors resulting in mutually costly outcomes, such as opponents in a game. Our results showed that by analyzing the interaction, the robot could better select interactive actions [3].

3.2 Partner Modeling

Several researchers have explored how humans develop mental models of robots (e.g. [20]. A mental model is a term used to describe a person's concept of how something in the world works [21]. We use the term partner model (denoted m^{-i}) to describe a robot's mental model of its interactive human partner. We use the term self model (denoted m^{i}) to describe the robot's mental model of itself. Again, the superscript -*i* is used to express individual *i*'s partner [11].

In prior work, Wagner presented an interact-and-update algorithm for populating outcome matrices and for creating increasingly accurate models of the robot's interactive partner [1]. The interact-and-update algorithm constructed a model of the robot's partner consisting of three types of information:

1) a set of partner features $\left(f_1^{-i},\ldots,f_n^{-i}\right)$;
2) an action model, A^{-i}; and
3) a utility function u^{-i}.

We use the notation $m^{-i}.A^{-i}$ and $m^{-i}.u^{-i}$ to denote the action model and utility function within a partner model. The dot is used to convey that the action model and utility function are contained within an overall partner model m^{i}. Wagner used partner features for partner recognition. Partner features, such as hair color, height, and age, allow the robot to recognize the partner in subsequent interactions. The partner's action model contained a list of actions available to that individual. The partner's utility function included information about the outcomes obtained by the partner when the robot and the partner select a pair of actions. Wagner showed that the algorithm could produce increasingly accurate partner models which, in turn, resulted in accurate outcome matrices. The results were, however, limited to static, not dynamic, models of the partner.

The self model also contains an action model and a utility function. The action model contains a list of actions available to the robot. Similarly the robot's utility function includes information about the robot's outcomes.

4. Deceptive Interaction

This article explores deceptive interaction. We investigate deceptive interaction with respect to two individuals—the mark and the deceiver. It is important to recognize that the deceiver and the mark face different problems and have different information. The mark simply selects the action that it believes will maximize its own outcome, based on all of the information that it has accumulated. The deceiver, on the other hand, acts in accordance with Bond and Robinson's definition of deception, providing a false communication for its own benefit [5]. With respect to our running example, the robot acts as the deceiver—providing false information as to its whereabouts. The mark then is the enemy soldier searching for the robot. We will assume henceforth that the deceiver provides false communication through the performance of some action in the environment. The sections that follow begin by examining the phenomena of deception, how to decide when to deceive, and finally provide a method for deciding how to deceive.

4.1 The Phenomena of Deception

Bond and Robinson's definition of deception implies the following five steps:

1. The deceiver selects a false communication to transmit.
2. The deceiver transmits the information contained within the false communication.
3. The information is received by the mark.
4. The mark interprets the information.
5. The interpreted information influences the mark's selection of actions.

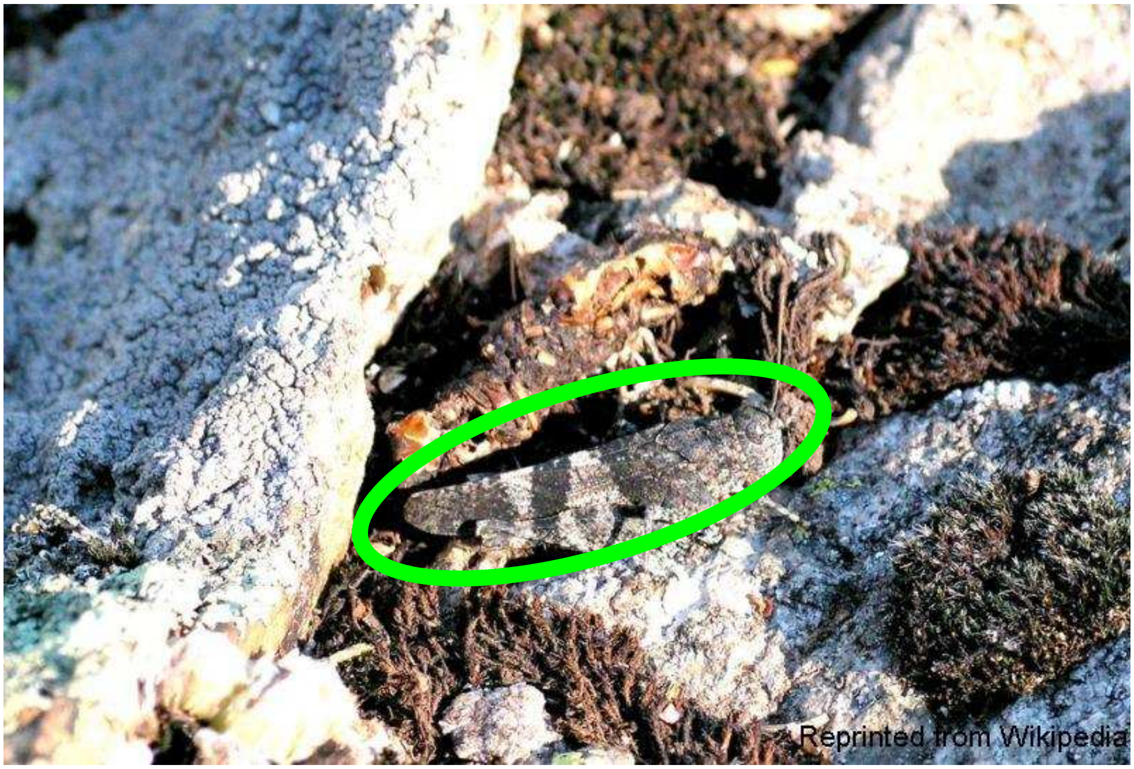
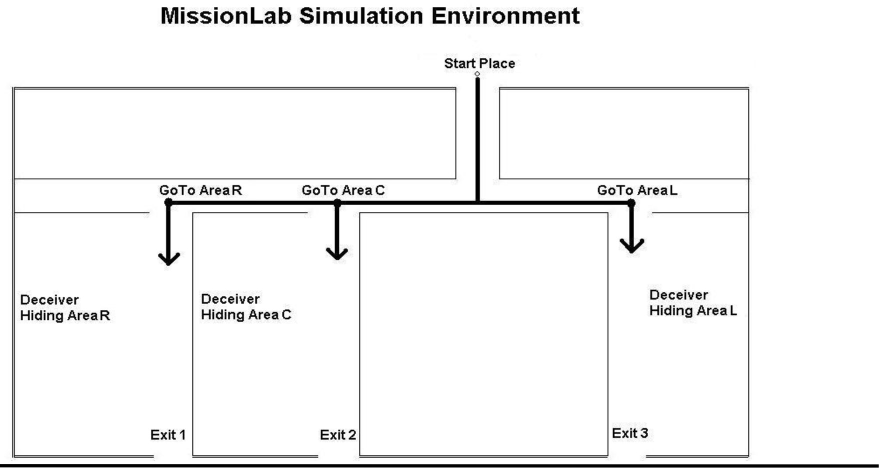


Figure 2 A grasshopper uses camouflage to deceive potential predators. In this case, color and shaping transmitted by the grasshopper falsely communicate the presence of rock and moss masking the presence of the grasshopper.

Consider, for instance, the use of camouflage by the grasshopper in Figure 2. First, the grasshopper's biological processes produce a false communication in the form of color changes to the grasshopper's body. The grasshopper's change in color is transmitted visually to its surroundings. Predators receive the visual information and interpret the information to indicate that no grasshopper exists at that location. This interpretation inhibits the predator from selecting predation behaviors.



True Matrix

Deceiver's true matrix

		Deceiver
		$a_1^D =$ GoLeft
Mark	$a_1^M =$ GoLeft	-10 10
	$a_2^M =$ GoCenter	10 -10
	$a_3^M =$ GoRight	10 -10

Deceiver's matrix representing mark's decision problem

		Deceiver		
		$a_1^D =$ GoLeft	$a_2^D =$ GoCenter	$a_3^D =$ GoRight
Mark	$a_1^M =$ GoLeft	-10 10	10 -10	10 -10
	$a_2^M =$ GoCenter	10 -10	10 -10	10 -10
	$a_3^M =$ GoRight	10 -10	10 -10	10 -10

Induced Matrix

Deceiver's matrix representing mark's decision problem after the false communication.

		Deceiver		
		$a_1^D =$ GoLeft	$a_2^D =$ GoCenter	$a_3^D =$ GoRight
Mark	$a_1^M =$ GoLeft	10 -10	10 -10	10 -10
	$a_2^M =$ GoCenter	10 -10	-10 10	10 -10
	$a_3^M =$ GoRight	10 -10	10 -10	10 -10

Figure 3 An example environment, true matrix, and induced matrix is pictured above. The environment contains three corridors in which the deceiver can hide. The true matrix reflects the deceiver's knowledge of the action it intends to select. In the true matrix on the left the deceiver has randomly selected the `GoLeft` action. The matrix depicts the deceiver's outcomes and their dependence on the mark's action. The true matrix to the right depicts the decision problem faced by the mark with the outcomes that would result given the action selected by the deceiver. The induced matrix depicts the mark's decision problem after the false communication. The outcome values ±10 were arbitrarily chosen.

Outcome matrices can be used to reason about the situation faced by the deceiver and the mark. Let a_1^D, a_2^D, a_3^D and a_1^M, a_2^M, a_3^M represent generic actions possessed by the deceiver and the mark respectively. We use the term *true matrix* to describe the outcome matrix representing the actual outcome obtained by both the mark and the deceiver had the false communication not occurred. From a game theoretic point of view, this is a situation of asymmetric information [22]. With respect to our running example, the true matrix represents the different outcome patterns resulting when the robot and enemy select hide and search actions. Figure 3 depicts this scenario assuming that the value in terms of utility of locating the deceiver is +10 and -10 if the deceiver does not locate the mark. A key facet of deception is the fact that the deceiver knows the true matrix but the mark does not. Consider, for instance the true matrix resulting from the deceiver's decision to hide in the left corridor. The true matrix on the left side of Figure 3 depicts the matrix from the deceiver's perspective. The true matrix on the right side of Figure 3 depicts the deceiver's understanding of the decision problem faced by mark. It includes the true outcome values that the mark will receive by choosing to search the center or right corridor. The deceiver's task is to provide information or to act in a way that will influence the mark to select a_2^M =`GoCenter`, a_3^M =`GoRight` rather than a_1^M =`GoLeft`. To do this, the deceiver must convince the mark that 1) the selection of a_1^M =`GoLeft` is less beneficial then it actually is; 2) the selection of a_2^M =`GoCenter`, a_3^M =`GoRight` is more beneficial then is actually is or 3) both.

The deceiver accomplishes this task by providing a false communication, i.e. a set of tracks leading elsewhere. This communication is false because it conveys information which falsely reflects the outcome of a particular action choice. The false communication results in another matrix which we term the *induced* matrix (Figure 3 bottom). The induced matrix represents the situation that the false communication has been led the mark to believe is true. In our running example, the hiding robot might create muddy tracks leading up to the center corridor (the false communication) while in fact the robot is actually hiding in the left corridor.

The preceding discussion has detailed several basic interactive situations and concepts underlying deception. Numerous challenges still confront the deceiver. The deceiver must be able to decide **if** a situation justifies deception. The deceiver must also be capable of developing or selecting a strategy that will communicate the **right** misleading information to induce the desired matrix upon the mark. For instance, a robot capable of deceiving the enemy as to its whereabouts must first be capable of recognizing that the situation demands deception. Otherwise its deception strategies are useless. In the sections that follow, we first develop a method that allows the robot to determine if deception is necessary. Afterward, we detail a method for how to deceive.

4.2 Deciding when to Deceive[1]

Recognizing if a situation warrants deception is clearly of importance. Although some application domains (such as covert operations) might demand a robot which simply deceives constantly and many other domains will demand a robot which will never deceive, this article focuses on robots which will occasionally need to deceive. The problem then for the robot, and the purpose of this section, is to determine on which occasions the robot should deceive.

[1] Portions of this discussion have also appeared in (Wagner & Arkin, 2009) © 2009 IEEE.

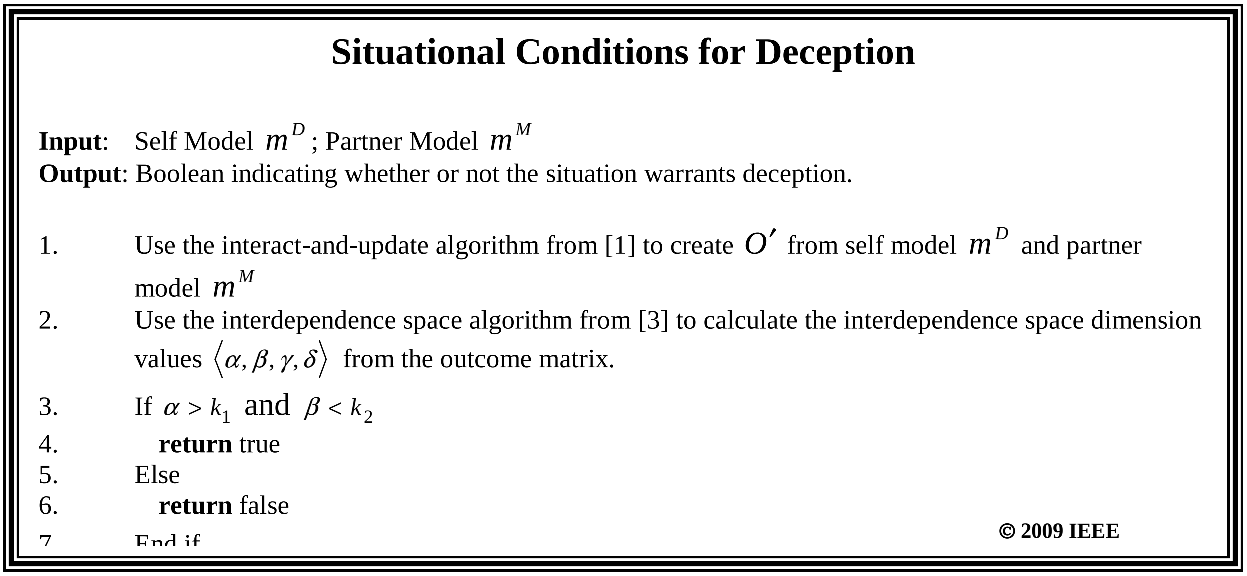
Section 3.1 detailed the use of outcome matrices as a representation for interaction and social situations. As described in that section, social situations represent a generic class of interactions. We can then ask what type of social situations justifies the use of deception? Our answer to this question will be with respect to the dimensions of the interdependence space. Recall from section 3.1 that the interdependence space is a four-dimensional space describing all possible social situations (see [3] for a depiction of the interdependence space). Posed with respect to the interdependence space, our task then becomes to determine which areas of this space describe situations that warrant the use of deception and to develop and test an algorithm that tests whether or not a particular interaction warrants deception.

As before, Bond and Robinson's definition of deception, providing a false communication for one's own benefit, will serve as our starting place [5]. With respect to the task of deciding when to deceive there are two key conditions in the definition of deception. First, the deceiver provides a **false** communication and second that the deceiver receives a **benefit** from this action. The fact that the communication is false implies conflict between the deceiver and the mark. If the deceiver and the mark had corresponding outcomes a true communication could be expected to benefit both individuals. The fact that the communication is false demonstrates that the deceiver cannot be expected to benefit from communications which will aid the mark. In our running example, a robot that leaves tracks leading to its actual hiding position is not deceiving because it is providing a true communication. On the other hand, all signals leading the mark away from the robot's hiding place will benefit the robot and not benefit the mark.

This figure is provided as a separate attachment

Figure 4 A two dimensional representation of the interdependence space showing the correspondence dimension (X) and the interdependence dimension (Y) is presented above. Areas of low interdependence (independent outcomes bottom half of graph) tend not to warrant deception because the actions of the mark will have little impact on the deceiver. Similarly, areas of correspondence (right portion of the graph) do not require false communication as actions beneficial for the mark are also beneficial for the deceiver. It is only the top left of the graph, representing areas in which the deceiver depends on the actions of the mark and is also in conflict with the mark, in which deception is warranted.

The second condition requires that the deceiver receive a benefit from the deception. This condition implies that the deceiver's outcomes are contingent on the actions of the mark. With respect to the interdependence space this condition states that the deceiver is dependent upon the actions of the mark. In other words, this is a situation of high interdependence for the deceiver. If this condition were not the case, then the deceiver would receive little or no benefit from the deception. Again, relating back to our running example, if the robot does not gain anything by hiding from the soldiers then there is no reason for deception. Figure 4 depicts a subspace of the interdependence space with respect to the two dimensions critical for deception.



Situational Conditions for Deception

Input: Self Model m^D; Partner Model m^M
Output: Boolean indicating whether or not the situation warrants deception.

```
Use the interact-and-update algorithm from [1] to create $O'$ from self model $m^D$ and partner model $m^M$
Use the interdependence space algorithm from [3] to calculate the interdependence space dimension values $\langle \alpha, \beta, \gamma, \delta \rangle$ from the outcome matrix.
If $\alpha > k_1$ and $\beta < k_2$
   return true
Else
   return false
End if
```

© 2009 IEEE

Figure 5 An algorithm for determining whether or not a situation warrants deception is presented above [23]. The algorithm takes as input the robot's self model and partner model. It uses the interact-and-update algorithm from [1] to produce an expected outcome matrix for the situation, O'. Next the interdependence space algorithm from [3] is used to generate the interdependence space dimension values $\langle \alpha, \beta, \gamma, \delta \rangle$ for the situation. Finally, if the value for interdependence is greater then some application specific constant k_1 and the value for correspondence less than some application specific constant k_2, the situation warrants deception.

Given the description above, we can construct an algorithm for deciding when to deceive (Figure 5). The aim is to determine if a situation warrants the use of deception. The algorithm draws heavily from our previous work in the area of human-robot interaction [1, 3]. The input to the algorithm is the robot's model of itself and of its interactive partner. These models are used in conjunction with Wagner's interact-and-update algorithm to produce an outcome matrix O', i.e., the true matrix (example in Figure 2) [1]. In the second step, the interdependence space mapping algorithm is used to calculate the situation's location in the interdependence space [3]. If the situation's location in the interdependence space indicates sufficient interdependence ($\alpha > k_1$) and conflict ($\beta < k_2$) then the situation can be said to warrant deception.

For robots, these conditions comprise necessary but not sufficient conditions for deception. Sufficiency also demands that the robot is capable of producing a false communication which will influence the mark in a manner beneficial to the deceiver. In order for this to be the case, the deceiver must have the ability to deceive. The presence or absence of the ability to deceive rests upon the deceiver's action set. Section 4.4 explores the robot's decision of how to deceive.

We contend the algorithm in Figure 5 allows a robot to recognize when deception is justified. The following two sections test this hypothesis first qualitatively and quantitatively.

4.2.1 Qualitative Comparison of Situational Conditions

In this section we qualitatively compare examples of those situations which meet the conditions for deception expounded in the previous section from those which do not. Our goal is to demonstrate that the algorithm in Figure 5 meets the same situational conditions which intuitively reflect those situations that humans use deception. Additionally, we strive to show that situations in which humans rarely, if ever, use deception are also deemed not to warrant deception by our algorithm. The purpose of this analysis is to provide support for the hypothesis that the algorithm in Figure 5 does relate to the conditions underlying normative interpersonal deception. It is challenging, if not impossible, to show conclusively outside of a psychological setting that indeed our algorithm equates to normal human deception processes.

Table **1** lists 5 different game/interdependence theoretic social situations. Each situation was used as the matrix O' from the first step of our algorithm for the situational conditions for deception. The values for constants were $k_1 = 0.66$ and $k_2 = -0.33$. The rightmost column states whether or not the algorithm indicates that the situation warrants deception.

Table 1 Qualitative exploration of the deception characteristics of several important social situations. Several situations, such as the Competitive situation and the Prisoner's dilemma, indicate that these situations warrant deception. Others, such as the Cooperative situation and the Trust situation, do not warrant deception. Finally, the Chicken situation stands as a border case which depends on the particular values in the matrix and constants associated with the algorithm.

Social Situations				
Name	**Verbal Description** (based on [19]	**Example Outcome Matrix**	**Interdependence Space Location**	**Situational Deception?**
Cooperative Situation	Each individual receives maximal outcome by cooperating with the other individual.	$\begin{array}{\|cc\|cc\|}\hline & 12 & & 6 \\ 12 & & 6 & \\ \hline & 6 & & 0 \\ 6 & & 0 & \\ \hline\end{array}$	0.5, 1.0, -0.5, 0.0	No
Competitive Situation	Each individual gains from the other individual's loss. Maximal outcome is gained through non-cooperation.	$\begin{array}{\|cc\|cc\|}\hline & 6 & & 12 \\ 6 & & 0 & \\ \hline & 0 & & 6 \\ 12 & & 6 & \\ \hline\end{array}$	0.5, -1.0, -0.5, 0.0	Yes
Trust Situation	In this situation, cooperation is in the best interests of each individual. If, however, one individual suspects that the other will not cooperate, non-cooperation is preferred.	$\begin{array}{\|cc\|cc\|}\hline & 12 & & 8 \\ 12 & & 0 & \\ \hline & 0 & & 4 \\ 8 & & 4 & \\ \hline\end{array}$	1.0, 0.2, -0.3, 0.0	No
Prisoner's Dilemma Situation	Both individuals are best off if they act non-cooperatively and their partner acts cooperatively. Cooperation and non-cooperation, results in intermediate outcomes.	$\begin{array}{\|cc\|cc\|}\hline & 8 & & 12 \\ 8 & & 0 & \\ \hline & 0 & & 4 \\ 12 & & 4 & \\ \hline\end{array}$	0.8, -0.8, -0.6, 0.0	Yes
Chicken Situation	Each individual chooses between safe actions with middling outcomes and risky actions with extreme outcomes.	$\begin{array}{\|cc\|cc\|}\hline & 8 & & 12 \\ 8 & & 4 & \\ \hline & 4 & & 0 \\ 12 & & 0 & \\ \hline\end{array}$	1.0, 0.2, -0.3, 0.0	Yes/No **© 2009 IEEE**

To give an example of how the results were produced consider the first situation in the table, the Cooperative Situation. A representative outcome matrix for the situation is used as the matrix O' from the first step of the algorithm. Next, in the second step of the algorithm the values for the fourth column of the table are calculated: the interdependence space dimension values. For the Cooperative Situation these values are $\{0.5, 1.0, -0.5, 0\}$. Because $\alpha < 0.66$ and $\beta > -0.33$ the algorithm returns false, indicating the situation does not warrant deception. The following situations were analyzed:

- The Cooperative situation describes a social situation in which both individuals interact cooperatively in order to receive maximal outcomes. Although often encountered in normative interpersonal interactions, because the outcomes for both individuals correspond, these situations seldom involve deception. For example, deception among teammates is rarely employed as it is counter to the dyad's mutual goals.
- In contrast to the Cooperative Situation, the Competitive situation does warrant the use of deception. This situation is again an example of a *k*-sum game in which gains by one individual are losses for the other individual. Hence, deception in interpersonal Competitive situations is common. Deception among competitors, for example, is extremely common and some games, such as poker, are even founded on this principle.
- The Trust Situation describes a situation in which mutual cooperation is in the best interests of both individuals. Yet, if one individual does not cooperate then mutual non-cooperation is in both individuals best interest. Interpersonal examples of Trust Situations include lending a friend money or a valuable asset. This situation does not demand deception because again both individuals' mutual interests are aligned.

- The Prisoner's Dilemma is perhaps the most extensively studied of all social situations [17]. In this situation, both individuals depend upon one another and are also in conflict. These conditions make the Prisoner's Dilemma a strong candidate for deception. It is in both individuals best interest to influence that action selection of the other individual. As detailed by Axelrod, Prisoner's Dilemma situations include military and police enforcement situations involving actual interpersonal interaction that often do entail deception [17].
- The Chicken situation is a prototypical social situation encountered by people. In this situation each interacting individual chooses between safe actions with intermediate outcomes or more risky actions with more middling outcomes. An example might be the negotiation of a contract for a home or some other purchase. Whether or not this situation warrants deception depends on the relative outcome value of the safe actions compared to the risky actions. If the value of the risky action is significantly greater then the value of the safe actions then deception will be warranted.

Table **1** and the analysis that followed examined several situations and employed our situational conditions for deception algorithm to determine if the conditions for deception were met. In several situations our algorithm indicated that the conditions for deception were met. In others, it indicated that these conditions were not met. We related these situations back to interpersonal situations commonly encountered by people, trying to highlight the qualitative reasons that our conditions match situations involving people. Overall, this analysis provides preliminary evidence that our algorithm does select many of the same situations for deception that are selected by people. While much more psychologically valid evidence will be required to strongly confirm this hypothesis, the evidence in this section provides some support for our hypothesis.

4.2.2 Quantitative Examination of Situational Conditions Warranting Deception

In this section we examine the hypothesis that by recognizing situations which warrant deception, a robot is afforded advantages in terms of outcome obtained. Specifically, a robot that can recognize that a situation warrants deception can then choose to deceive and thereby receive more outcome overall, than a robot which does not recognize that a situation warrants deception. Although this experiment does not serve as evidence indicating that our situational conditions for deception relate to normative human conditions for deception, it does show that robots which recognize the need for deception have advantages in terms of outcome received when compared to robots which do not recognize the need for deception.

At first glance this experiment may appear trivial given the definition of deception. There are, however, several reasons that the study is important. First, we do not know the magnitude of the benefit resulting from deception. Does the capacity to deceive result in significantly greater benefit over an individual that does not deceive? Similarly, how often must one deceive in order to realize this benefit? Second, we do not know how this benefit is affected by unsuccessful deception. Is the benefit realized by 80% successful deception the same as 100% successful deception? Finally, this definition was developed for biological systems. Hence, we need to verify that artificial systems such as agents and robots will likely realize the same benefit as a biological system. In other words, we need to verify that the benefit is not something unique to biological systems. While the answers to these questions may seem straightforward, they are an important starting place given that this paper lays the foundation for a largely unexplored area of robotics.

We conducted a numerical simulation to estimate the outcome advantage that would be afforded to a robot that used the algorithm in Figure 5 versus a robot which did not. Our numerical simulation of interaction focuses on the quantitative results of the algorithms and processes under examination and does not attempt to simulate aspects of the robot, the human, or the environment. As such, this technique offers advantages and disadvantages as a means for discovery. One advantage of a numerical simulation experiment is that a proposed algorithm can be tested on thousands of outcome matrices represent thousands of social situations. One disadvantage is that, because it is not tied to a particular robot, robot's actions, human, human's actions, or environment, the results, while extremely general, have not been shown to be true for any existent social situation, robot, or human. The experiment involved two simulated robots. Each selected nominal actions from their outcome matrices and received the resulting values, but no actual actions were performed by either individual.

These simulations involved the creation of 1000 outcome matrices populated with random values. Artificial agents abstractly representing robots selected actions based on the outcome values within the matrices. These outcome matrices were also abstract in the sense that the rewards and costs are associated by selecting one of two non-specific actions. Symbolic placeholders such as a_1 and a_2 are used in place of actual actions. The actions are grounded in the rewards and costs that the robot expects them to produce. This may be the only practical way to examine thousands of situations at a time and to draw general conclusions about the nature of deception itself outside of one or two specific situations. Both the deceiver and the mark selected the action which maximized their respective outcomes. Figure 6 depicts the experimental procedure with an example.

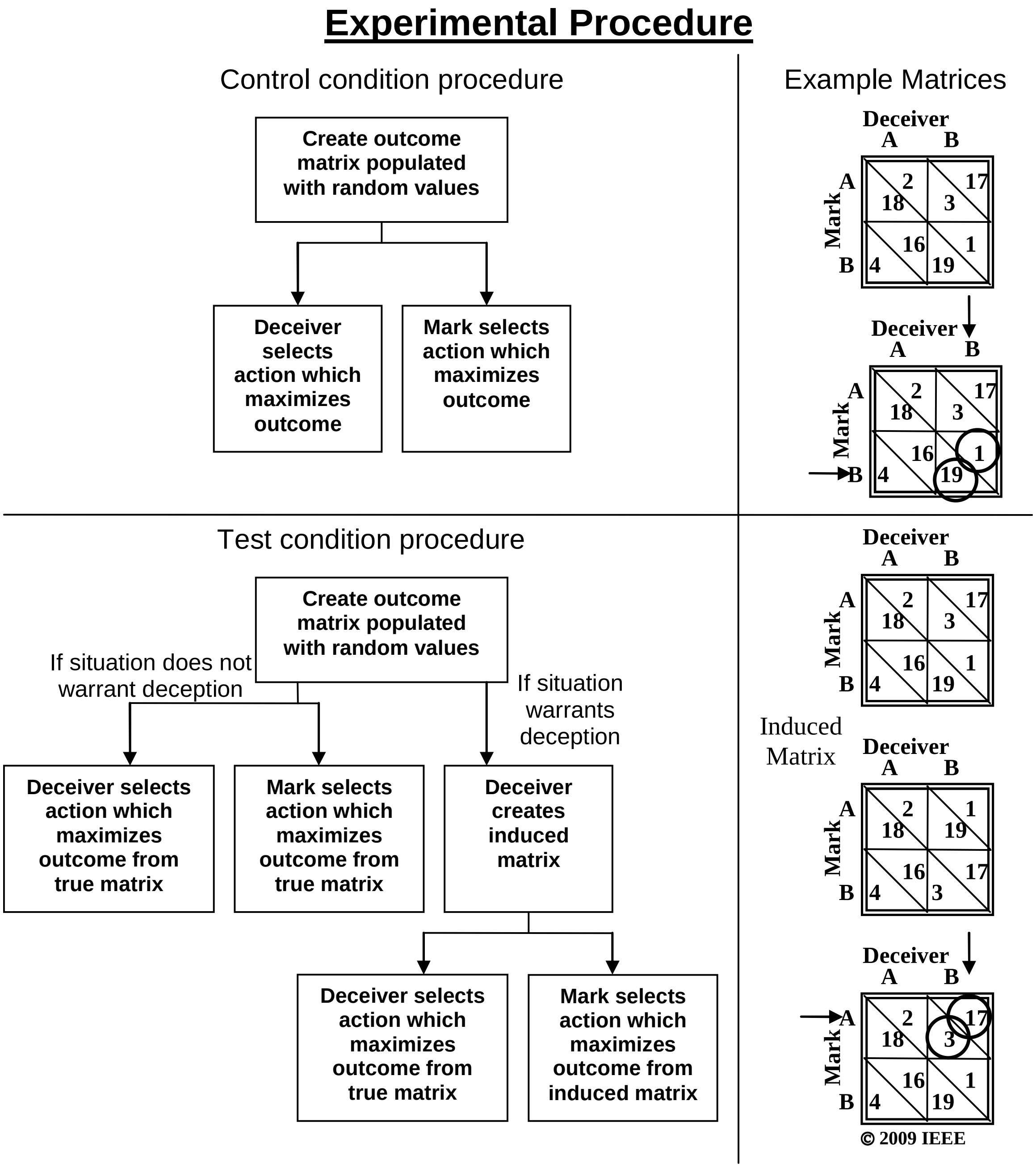

Experimental Procedure
Control condition procedure
Create outcome matrix populated with random values
Deceiver selects action which maximizes outcome
Mark selects action which maximizes outcome
Test condition procedure
Create outcome matrix populated with random values
If situation does not warrant deception
If situation warrants deception
Deceiver selects action which maximizes outcome from true matrix
Mark selects action which maximizes outcome from true matrix
Deceiver creates induced matrix
Deceiver selects action which maximizes outcome from true matrix
Mark selects action which maximizes outcome from induced matrix
Example Matrices
Deceiver
A
B
Mark
A
B
2
18
17
3
16
4
1
19
Induced Matrix
2
18
1
19
16
4
17
3
© 2009 IEEE

Figure 6 The experimental procedure is depicted above. In the control condition, random outcome matrices are created and actions are selected from these matrices. In the test condition, if the situation warrants deception then deceiver creates an induced matrix which the mark selects an action from. Example matrices are depicted on the right hand side of the figure.

Three experimental conditions were examined. The first condition was a control condition devoid of deception. In this condition both the deceiver and the mark simply selected the action which maximized their individual outcomes. This condition represents the null hypothesis in that if performance in the control is as great or greater then performance using our algorithm then the recognition of the situational conditions for deception via our algorithm offers no benefit to the agent.

In the two experimental conditions, the deceiver used the algorithm from Figure 5 to determine if the outcome matrix warranted deception. If it did, then the deceiver produced an induced matrix which was used by the mark to select an action while the deceiver selected an action based on the true matrix. In the perfect deception condition the mark always selected an action based on the induced matrix. In the 80% deception condition, the mark selected an action from the induced matrix 80% of the time and from the true matrix 20% of the time. The importance of the 80% percent deception condition is that it indicates how quickly the benefit of deception decreases with an imperfect deception strategy.

The independent variable was the condition: no deception, perfect deception, or 80% successful deception. The dependent variable was the amount of outcome received by each simulated agent.

Relating back to our running example, in both the control and the test conditions, the deceiver interacts in thousands of situations at the military base. Most of these situations do not warrant deception and hence the control and test robots act the same. Only the robots in the experimental condition which are using our algorithm, however, recognize the situations that do warrant deception. In this case these experimental robots use a deceptive strategy, such as creating a false trail to hide, to create an induced matrix that influences the behavior of the mark. The deceiving robot then hides in a different location.

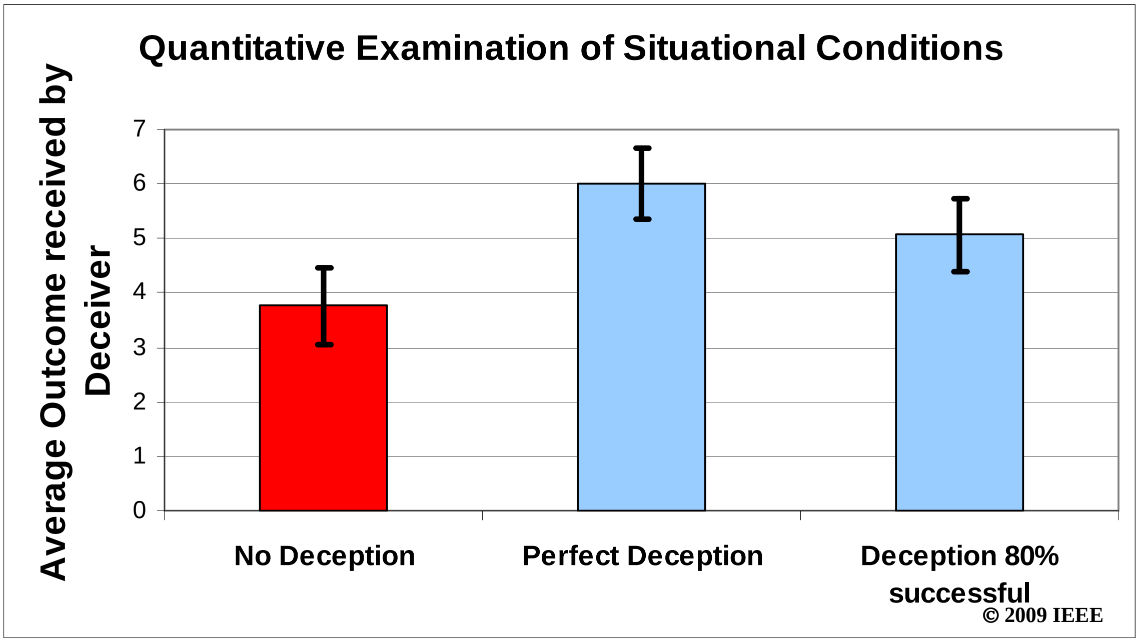


Figure 7 Experimental results from our investigation of the situational conditions warranting deception. The perfect deception and 80% successful deception conditions result in significantly ($p < 0.01$) greater outcome than the no deception condition. This result indicates that an agent or robot that can recognize and act upon the situational conditions for deception will be better able to choose the best action.

Figure 7 presents the results. The recognition and use of deception results in significantly more outcome ($p < 0.01$ two-tailed no deception versus perfect deception and no deception versus 80% successful deception) than not recognizing and using deception. Of the 1000 random situations the simulated agents faced, 19.1% met the conditions for deception. Hence, all of the difference in outcome among the various conditions resulted from better action selection on the part of the deceiver in only 191 situations. This experiment serves as evidence that an artificial agent or robot that can recognize and react to situations which warrant the use of deception will be significantly better suited to maximize their outcomes and hence their task performance.

These results are important in that they demonstrate that:

1) That a robot or agent that recognizes when to deceive will obtain significantly more outcome than a robot that does not;
2) most of the difference results from a relatively small (19.1) percentage of situations;
3) imperfect deception does impact the amount of outcome obtained; and
4) Bond and Robinson's biological definition for deception can be used in conjunction with an interdependence theory framework to develop methods for robots to recognize when deception is warranted.

Still, the experiments presented in this section have assumed that the robot is capable of deception. Clearly this assumption is not as yet justified. In the next section we investigate the method that the deceiver uses to effectively deceive the mark.

4.3 Deciding how to Deceive

Bond and Robinson's definition of deception implies a temporal order [5]. The deceiver must provide a false communication before the mark has acted. A false communication provided after the mark has acted cannot be expected to benefit the deceiver. Several authors have recognized the need for a particular temporal order during deceptive interactions [7, 10]. Gerwehr and Glenn detail a planning process necessary for deception with respect to the military domain [7]. Ettinger and Jehiel provide a theoretical basis for a prototypical pattern of interactions relating to deception [10]. Floreano et al., on the other hand, demonstrate that deception can occur with little or no actual planning while still confirming that the temporal order described above must be preserved [13]. With respect to our running example, the robot cannot leave muddy tracks up to the first hiding place after the enemy soldier has already selected an area to search. The deceptive action must be completed beforehand.

Our algorithm for acting deceptively is structured with this temporal order in mind. It consists of four stages. First the deceiver determines if the situation does indeed warrant the use of deception. Next, the deceiver creates the induced matrix. Recall from section 4.1, the induced matrix is the matrix that the deceiver wishes the mark to believe. Next, the deceiver selects the best false communication to convince the mark that the induced matrix is the true matrix. Finally, the deceiver and the mark perform their actions in the environment.

Acting Deceptively

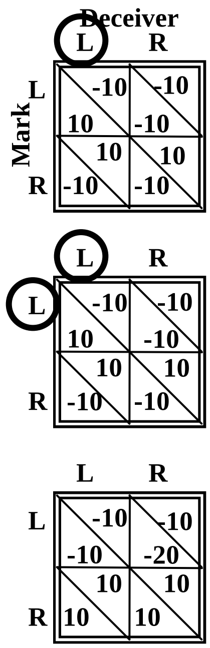
```
Input:   Partner Model $m^{-i}$; true matrix $O'$; constant $k_1$, $k_2$
Output: None

1.  Check if the situation warrants deception, if so then continue
//Calculate the induced matrix
2.  Set $a^{\min} \in A^{-i}$ such that $O'(a^i, a^{\min}) = \min(o^i)$      //find the mark's action which will
                                                                         //minimize the deceiver's outcome
3.  $\tilde{O}(a^{\min}) = O'(a^{\min}) - k_1$      //Subtract $k_1$ from the mark's outcome for action $a^{\min}$
4.  $\tilde{O}(a^{-i\neq\min}) = O'(a^{-i\neq\min}) + k_2$      //Add $k_2$ from the mark's outcome for all other
                                                          //actions producing the induced matrix
//Select the best false communication
5.  for each $\gamma_j \in \Gamma$      //for each potential false communication
6.      $g(m^{-i}, \gamma_j) = m^{-i*}$      //calculate the change the comm. will have on the partner model
7.      $f(m^i, m^{-i*}) = O^*$      //calculate the resulting matrix from the new partner model
8.      if $O^* \approx \tilde{O}$      //if the matrix resulting from the false comm. is approx. equal to
                                   //the matrix we wish to induce, then
9.          Set $\gamma^* = \gamma_j$      //set the best communication to the current communication
//Interact
10. Deceiver produces false communication $\gamma^* \in \Gamma$, the signal resulting in maximum outcome.
11. Deceiver uses matrix $O'$ to select action $a^D \in A^D$ which maximizes deceiver's outcome.
12. Mark produces induced matrix $\hat{O}$.
13. Mark selects action from induced matrix $\hat{O}$.
```

Figure 8 An algorithm for acting deceptively. The algorithm takes as input the deceiver's model of the mark, the true matrix and two constants related to the deceiver's strategy for fooling the mark.

The algorithm begins by checking if the situation warrants deception. If so, then the deceiver attempts to determine what the characteristics of the induced matrix will be. Recall from section 4.1 that the deceiver can either try to increase the probability that the mark will select an action favorable to the deceiver, decrease the probability that the mark will select an action unfavorable to the deceiver, or both. In our algorithm, control of the constants $k_1 \geq 0$ and $k_2 \geq 0$ allows the deceiver to both decrease the outcome of an unfavorable action being selected (k_1) and increase the outcome of a favorable action being selected (k_2). The upper bounds for the values of k_1 and k_2 are limited by the deceiver’s set of false communications. Greater values for the constants will result in an induced matrix which is not achievable given the deceiver’s set of false communications. The actual values for the constants can potentially be derived from previous experience with similar situations, the robot’s motivations, or from the true matrix itself.

Creating the Induced matrix in a 2x2 Hide-and-seek example



True matrix: deceiver has decided to hide in the left corridor

L is the mark’s action that would minimize the deceiver’s outcome. Thus $a^{\min} = L$

The induced matrix: a new matrix created by subtracting $k_1 = 10$ from L and adding $k_2 = 10$ to R.

Figure 9 An example of the creation of the induced matrix by the acting deceptively algorithm on a 2x2 hide-and-seek example.

Steps 2-4 create the induced matrix by reducing the outcome from the action deemed not favorable to the deceiver and adding outcome to the actions deemed favorable to the deceiver. The second step locates the mark’s action or actions that will result in a reduced amount of outcome being obtained by the deceiver. Next, in step 3, the deceiver subtracts a constant value (k_1) from the mark’s outcomes for the action or actions found in step 2. The value for this constant should be chosen such that the increase in outcome makes the alternative action or actions appear to be favorable with respect to the mark’s other action choices. The forth step adds a constant value (k_2) to each of the mark's actions which are favorable to the deceiver. The result is the production of an induced matrix which will persuade the mark to select the action which is most favorable to the deceiver. In our running example, the deceiver's most favored action would be for the mark to search in the location that the deceiver is not hiding. Hence, the induced matrix, in this case, makes searching in the location that the deceiver is hiding seem incorrect to the mark. Figure 9 presents an example of the creation of the induced matrix in 2x2 version of our hide-and-seek situation.

The next five steps of the algorithm attempt to determine which false communication would be the best communication to create the induced matrix within the mark. Intuitively, steps 5-9 iterate through the deceiver's set of possible false communications searching for the false communication that will produce an induced matrix which most closely resembles the induced matrix from step 3. To do this, in step 5 the algorithm iterates over all false communications. In step 6 the function $g(\cdot)$ calculates the impact of the false communication on the partner model. This function will be discussed in greater detail in the next section. Step 6 uses both the induced partner model and the deceiver's self model to create the matrix, O^*, that would be generated by this particular false communication. If the matrix induced by a particular false communication, O^*, is approximately equal to the desired induced matrix, $\tilde{O}$, then the false communication to be used, y_j , is saved.

Finally, in steps 10-13, the robot produces the false communication and selects an action from the true matrix O'. The mark reacts to the communication by generating its own internal matrix $\hat{O}$ which may or may not equal the induced matrix predicted by the deceiver. Finally, the mark selects an action from the matrix $\hat{O}$.

As presented, this algorithm makes a number of assumptions. We assume that the deceiver has a finite set of M false communications, $\Gamma = \{y_0, \ldots, y_M\}$, over which it is deliberating. This set of communications could more adequately be described as a set of deceitful actions with the purpose of providing false information to the mark. This set of deceitful actions could, potentially, be learned, or alternatively simply be given to the robot. The question of how the deceiver learns to act deceitfully remains open.

The seventh step assumes that a measure of similarity exists between outcome matrices. We empirically explored two different distance measures. Euclidean vector distance, $d = \sqrt{\sum_{i=1}^{N} \left(x1_i - x2_i\right)^2}$, was initially used as a measure of similarity. This measure, however, tends to emphasize the distance of each action pair equally without respect to how the distance impacts the mark's action selection. The underlying purpose of the comparison is to determine if the resulting matrix and the induced matrix will produce the same action selection by the mark. To get a better measure of similarity and dissimilarity with respect to the mark's action selection, we averaged each of the mark's actions across potential deceiver locations (averaging the rows of the matrix) to produce a vector, formally, $v_k = \frac{\sum_{j=1}^{N} O^M\left(a_k^M, a_j^D\right)}{N}$ where v_k is the average outcome value for the mark's *k*th action. We then used the Euclidean vector distance equation above to compare the matrix resulting from a potential false communication leading to the induced matrix. Other distance measures may also be possible.

4.3.1 Representing knowledge about the mark

The fifth step of the algorithm assumes the existence of a function $g(\cdot)$ that allows the deceiver to reason about and predict the impact of a false communication on the mark. The general question of how a piece of information will impact the state and beliefs of one's interactive partner is a challenging and open question. Arguably, this is also the most difficult step in normal interpersonal deception as it often requires detailed knowledge of the mark and can lead to infinite loops of common knowledge (i.e. the deceiver knows mud trails are fake, the mark knows that the deceiver knows that mud trails are fake, the deceiver knows that the mark know that the deceiver knows that mud trails are fake, and so on) [11]. Humans typically utilize knowledge about the other individual to determine how a false communication will affect the other individual's decision [24].

We use a Bayesian network to represent the deceiver's system of beliefs related to the mark. As mentioned in section 3, in game theory beliefs are used to probabilistically represent knowledge about oneself and one's partner [11]. A belief is formally [24] represented as the conditional probability $p_i(A|B)$, where $p_i(\cdot)$ is the likelihood function held by individual *i* that the random variable *A* takes a particular value given evidence in the form of a value for the random variable *B*.

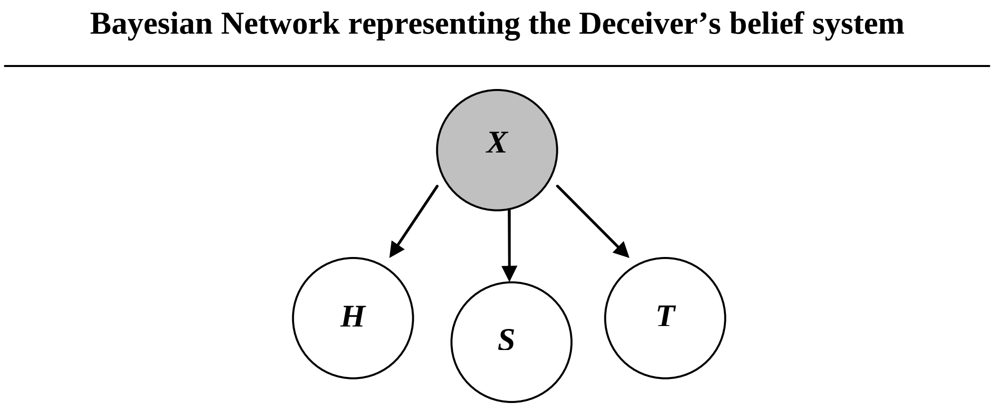


Figure 10 A Bayesian network representing the relationship of the random variables for the deceiver's belief system. The variable $X = \{l, r, c\}$ denotes the corridor the mark will choose to search in, $H = \{l, r, c\}$ denotes a heat signature, $S = \{l, r, c\}$ denotes the location of a sound, and $T = \{l, r, c\}$ denotes the location of tracks.

Four random variables captured the deceiver's system of beliefs related to the mark. For our example, the hidden variable $X = \{l, r, c\}$ denotes the location of the deceiver, the left (l), right (r), or center (c) corridor. The remaining variables are observable. The evidence variable $H = \{l, r, c\}$ denotes a heat signature located in the left (l), right (r), or center (c) corridor. The evidence variable $S = \{l, r, c\}$ denotes a sound signature with the same possible values as the random variable for a heat signature. The evidence variable $T = \{l, r, c\}$ denotes visible track signature again with the same potential values as the preceding variables. We assume that both the deceiver and the mark recognize the causal relationship between the random variables. That is, both individuals know that the deceiver's position causes the position of its heat signature, sound signature, and visible tracks and not vice versa. As shown by related work, causal relationships among the physical characteristics of objects can be learned by a robot [25]. Figure 10 depicts a directed graphical model representing the relationship among the different random variables. Given the model, the deceiver uses the probability $p_i(X \mid H, S, T)$ to predict the impact of possible false communications on the mark's model of the situation. For example, if the deceiver intends to use a heat signature placed in the left corridor and no other false communication $(H = l)$, then the term $p_D(X = l | H = l)$ represents the deceiver's belief that the mark will select the left corridor given a heat signature in the left corridor.

Predicting the impact of a false communication

Function $g(m^{-i}, \gamma_j) = m^{-i*}$

Input:

Partner model $m^{-i} = \left\{ \begin{array}{c} A^{-i} = \{l, r, c\} \\ U^{-i} = \left\{ \begin{array}{l} (l,l,10), (l,r,-10), (l,c,-10), \\ (r,l,-10), (r,r,10), (r,c,-10), \\ (c,l,-10), (c,r,-10), (c,c,10) \end{array} \right\} \end{array} \right.$

Evidence vector $\gamma_j = \langle H = r, S = r, T = r \rangle$

Junction tree algorithm: $p(X = l | H = r, S = r, T = r) = 0.0345$

$p(X = r | H = r, S = r, T = r) = 0.9310$

$p(X = c | H = r, S = r, T = r) = 0.0345$

Update utility values:

$u(l,l) = 0.0345 \times 10$, $u(l,r) = 0.9310 \times -10$, $u(l,c) = 0.0345 \times -10$,

$u(r,l) = 0.0345 \times -10$, $u(r,r) = 0.9310 \times 10$, $u(r,c) = 0.0345 \times -10$,

$u(c,l) = 0.0345 \times 10$, $u(c,r) = 0.9310 \times -10$, $u(c,c) = 0.0345 \times 10$

Return

$$m^{-i*} = \left\{ \begin{array}{c} A^{-i} = \{l, r, c\} \\ U^{-i} = \left\{ \begin{array}{l} (l,l,0.3), (l,r,-9.3), (l,c,-0.3), \\ (r,l,-0.3), (r,r,9.3), (r,c,-0.3), \\ (c,l,-0.3), (c,r,-9.3), (c,c,0.3) \end{array} \right\} \end{array} \right.$$

Figure 11 An example of the computations undertaken to predict the impact of a false communication. The figure details the calculations conducted as part of the function $g(\cdot)$ for the hide-and-seek example.

Our function $g(\cdot)$ uses the junction tree algorithm to perform statistical inference on the graphical model depicted in Figure 10 (see [26] for an overview). The vector $\mathrm{y}_{\mathrm{j}} = \langle H, S, T \rangle$ serves as input to the junction tree algorithm. The inference results in the posterior probability $p_i(X|H,S,T)$. The value of the posterior is then multiplied by the deceiver's belief of the mark's outcome values to produce revised outcome values given the evidence. These revised outcome values are stored in a revised partner model. Figure 11 details the computational process underlying our implementation of $g(\cdot)$.

4.3.2 Simulating deception in hide-and-seek

The study of deception and deception avoidance presents unique methodological challenges. Because the success or lack of success of a deception algorithm hinges not just on the deception algorithm itself, but also on the means of controlling the mark, deception results may not be indicative of successful deception per se, but rather of a weak mark. The challenge then becomes how to gauge the success of one's deception algorithm relative to a mark.

We utilize several techniques to deal with these challenges. First, all of the interactions between the deceiver and the mark are one-shot interactions. In other words, neither the deceiver nor the mark refine their initial models of the other based on prior experience. This prevents the experiment from degenerating into a competition of machine learning algorithms. Second, different types of marks were created each with different action selection strategies. This prevents a single strategy from acting as a dominant strategy for the deceiver. Finally, we do not compare the algorithm's results to a control algorithm. Because this is a new area of research without established metrics or ground truth, even a statistically significant increase in ability to deceive with respect to a control could simply be a reflection of a weak control.

Rather than attempting to empirically demonstrate the ability of our algorithm to deceive, our primary goal will be to use the algorithm as a tool to investigate the nature of deception itself. Specifically, we explore the relationship between a deceiver's knowledge of the mark and its ability to deceive. We hypothesize that additional knowledge about the mark aids the deceiver by allowing it to more accurately reason and predict the effect of a false communication will have on the mark.

This hypothesis may seem intuitive. To restate it, the more a priori knowledge that the deceiver has about the mark the better its ability to deceive should be. There are, however, numerous instances of deception which would seem to contradict this statement. Take, for example, the use of camouflage as a method of deception (Figure 2). Typically the deceiver has little, if any, explicit knowledge pertaining to the mark yet the use of camouflage works flawlessly. We argue that in these cases the deceiver has tacit and/or implicit knowledge related to the perceptual abilities of the deceiver [27]. Also consider the debate within the primatology community as to whether or not the use of deception is indicative of theory of mind [2]. The results of these experiments are valuable in that they demonstrate possible computational underpinnings by which theory of mind could influence one's ability to deceive.

We conducted both simulation and robot experiments to test this hypothesis. *MissionLab* was used to conduct our simulation experiments. *MissionLab* is a robot mission specification software suite which allows users to simulate multi-agent and multi-robot scenarios, design robot behaviors and instantiate missions using situated embodied robotic hardware [28].

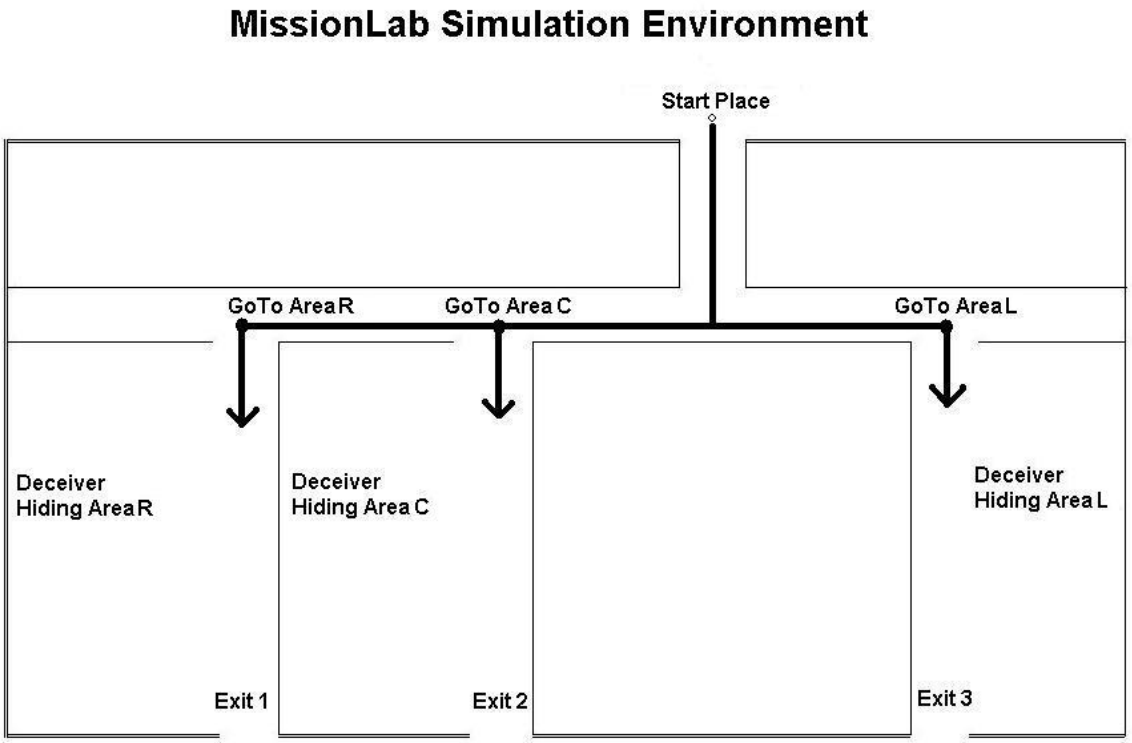


Figure 12 The multi-robot hide-and-seek simulation environment used for our experiments.

Our experiment involves multi-robot hide-and-seek [29]. Hide-and-seek is an agreeable paradigm for deception research because it is a transformation of the well-studied robot foraging problem [27]. In traditional robot foraging a robot searches an environment for attractors. The dependent variable is typically the number of attractors located or the mean time to locate a fixed number of attractors. Hide-and-seek varies the foraging problem by making the task multi-agent and placing the agents in conflict; the success of the seeker in inversely related to the success of the hider. In our variation of hide-and-seek, the hider/deceiver hides in one of three areas pictured in Figure 12. The seeker/mark must select only one location where to search. As a paradigm for the study of deception, one could potentially use hide-and-seek to explore both sides of the deception problem. Namely one could develop algorithms and theories for improving deception by focusing on the role of the hider/deceiver. Alternatively, one can explore deception recognition or reduction techniques by creating methods for the seeker/mark.

The hide-and-seek environment utilized for this research was designed to relate to this paper's running example and is meant to force the seeker to select a single search path (Figure 12). As discussed in section 3.2, knowledge of an individual's action space is a necessary prerequisite for the construction of an outcome matrix. In this case, the seeker's action space consists of going down one of the three corridors in search of the hiding robot. Formally, for this experiment the seeker's action space can be represented as $A^M = \{$`GoToAreaLeft, GoToAreaCenter, GoToAreaRight`$\}$. These actions were constructed in *MissionLab* from a series of waypoints allowing the robot to move through and search the different corridors. Likewise the hider must decide which corridor to hide in. Formally, the hider's action space can be represented as $A^D = \{$`HideInAreaLeft, HideInAreaCenter, HideInAreaRight`$\}$. The hider's actions were also constructed in *MissionLab* from a series of waypoints that allow the deceiving robot to move to one of the three hiding locations within a corridor.

Prior to the interaction the hider must select a false communication and location to aid in its deception. In this experiment, the deceiver used three types of diversions as false communications: heat signatures, sounds, and visible tracks. Notionally, heat signatures acted as false communications to seekers with infrared sensors, sounds acted as false communications to seekers with auditory sensing, and visible tracks were capable of fooling seekers with vision. The use of these particular types of diversions simply serves to ground the experiments in a setting and a story. They do not limit the applicability of the results as evidence for or against our hypothesis. Formally, for this experiment, the hider's false communication set can be represented as $\Gamma = \{$`tracks`, `heat_signature`, `sound`$\}$. Each false communication also has a location, left, right, or center.

Table 2 The table below lists the different types of seekers. The seeker's perception as limited by the presence or absence of vision, auditory, and infrared sensors.

Seeker Types

Number	Sensors	Deception signal sensitivity
1	None	insensitive
2	Infrared	mildly sensitive
3	Auditory	mildly sensitive
4	Vision	mildly sensitive
5	Infrared, auditory	largely sensitive
6	Infrared, vision	largely sensitive
7	Auditory, vision	largely sensitive
8	Vision, auditory, infrared	completely sensitive

Seekers come in several types (Table 2). The seeker's ability to detect the hider's false communication depends on the seeker's suite of sensors. Seekers with all possible combinations of sensors were created. Seekers were considered insensitive to the hider's deception signal if they had no ability to sense the signal, mildly sensitive if they had only a single type of sensor, largely sensitive if they had two sensors, and completely sensitive if they had all three sensors.

The dependent variable in these experiments was whether or not the seeker located the hider. Random selection of a search path by the seeker would be expected to locate the hider in ~33% of the runs. The independent variable was the amount of a priori knowledge the hider was given related to the seeker's perceptual capabilities. The hider's a priori knowledge of the seeker was quantized as notional questions the hider could ask the seeker. The hider could only ask one of the three questions listed in Table 3. The independent variable ranged from zero questions (no knowledge of the seeker's type) to three questions (full knowledge of the seeker's type). The question asked by the hider was selected at random and the deceiver did not ask the same question multiple times. Responses to all questions were yes or no. The seeker always responded honestly. The communication of this information was accomplished via the passage of byte strings among the robots. Speech synthesis and recognition were not used. The ability of the hider to ask questions of the seeker served as a mechanism for knowledge transfer from the seeker to the hider. In more realistic applications of this work the hider would, perhaps, infer visually that the seeker had a color camera, microphone, or infrared sensor. The actual development of visual capabilities, however, was far beyond the scope of this work. Thirty runs of the preceding experimental procedure were conducted for each of the values of the independent variable.

Table 3 List of the each of the questions the hider could ask the seeker in order to gain knowledge of the seeker's type.

Hider Questions		
Number	**Question**	**Response**
1	Do you have an infrared sensor?	Yes/No
2	Do you have an auditory sensor?	Yes/No
3	Do you have a vision sensor?	Yes/No

Outcome Matrix Representing Hide-and-Seek

Hider

Seeker

	$a_1^D =$ GoLeft	$a_2^D =$ GoCenter	$a_3^D =$ GoRight
$a_1^M =$ GoLeft	-10 10	10 -10	10 -10
$a_2^M =$ GoCenter	10 -10	-10 10	10 -10
$a_3^M =$ GoRight	10 -10	10 -10	-10 10

Figure 13 An outcome matrix representing the hide-and-seek game. The matrix arbitrarily values the seeker finding the hider with an outcome value of +10 and not finding the seeker with an outcome value -10. The reverse is true for the hider.

The following procedure was used:

Experimental Procedure

1) The seeker's type is randomly selected.

2) Both the hider and the seeker begin at the location labeled "Start Place" in Figure 12.

3) The hider sends the seeker questions pertaining to the seeker's type in accordance with the independent variable.

4) The seeker responds.

5) The hider creates an initial partner model representing the seeker with action set $A^M =$ {`GoToAreaLeft`, `GoToAreaCenter`, `GoToAreaRight`} and the utilities values depicted in Figure 13.

6) The hider randomly selects an action from the set $A^D =$ {`HideInAreaLeft`, `HideInAreaCenter`, `HideInAreaRight`}

7) The hider creates the true matrix with respect to its action selection from step 6). Figure 3 depicts the result for the action `HideInAreaLeft`.

8) The hider uses the partner model created in step 5) and the true matrix from 6) with constants $k_1 = 20, k_2 = 20$ as input to the algorithm from Figure 8.

9) The algorithm results in the selection of a false communication from $\Gamma = \{y_0, \ldots, y_M\}$. The hider produces the false signal.

10) The hider performs the action $A^D =$ {`HideInAreaLeft`, `HideInAreaCenter`, `HideInAreaRight`}.

11) The seeker, in accordance with its type, perceives or does not perceive the hider’s false communication signal.

12) The seeker selects a corridor and searches the corridor.

13) If the seeker selects the correct corridor, the hider’s deception is recorded as a failure. Otherwise, the hider’s deception is recorded as a success.

14) Steps 1) through 13) are repeated.

Recall that we hypothesized that the more a priori knowledge that the deceiver has about the mark the better its ability to deceive should be. Hence we expected that additional questions on the part of the deceiver would result in better deception and reduced detection on the part of the seeker. Figure 14 depicts the results. We see that the rate at which the deception succeeds increases from 78 percent to 95 percent as the number of questions asked by the deceiver increases from 0 to 3. Confidence intervals for the independent variable were ±5, ±5, ±4, and ±3 percent, respectively. Hence, we can conclude that the additional information gained by the deceiver aids in deception. The importance of this result will be discussed in the conclusions.

We can also breakdown the results with respect to the seeker’s sensitivity to deception. We claimed in Table 2 that seekers sensitivity to deception varied from insensitive to extremely sensitive. Figure 15 graphically depicts the deception success for each type of seeker. As expected deception success rate for the seeker without sensors is approximately 66 percent. The deception success rate for mildly sensitive seekers is significantly greater than the success rate for insensitive seekers ($p < 0.01$ for all two-tailed). No significance difference resulted when comparing mildly sensitive seekers to largely sensitive seekers. The trend line depicts the general increase in deception success with seeker sensitivity.

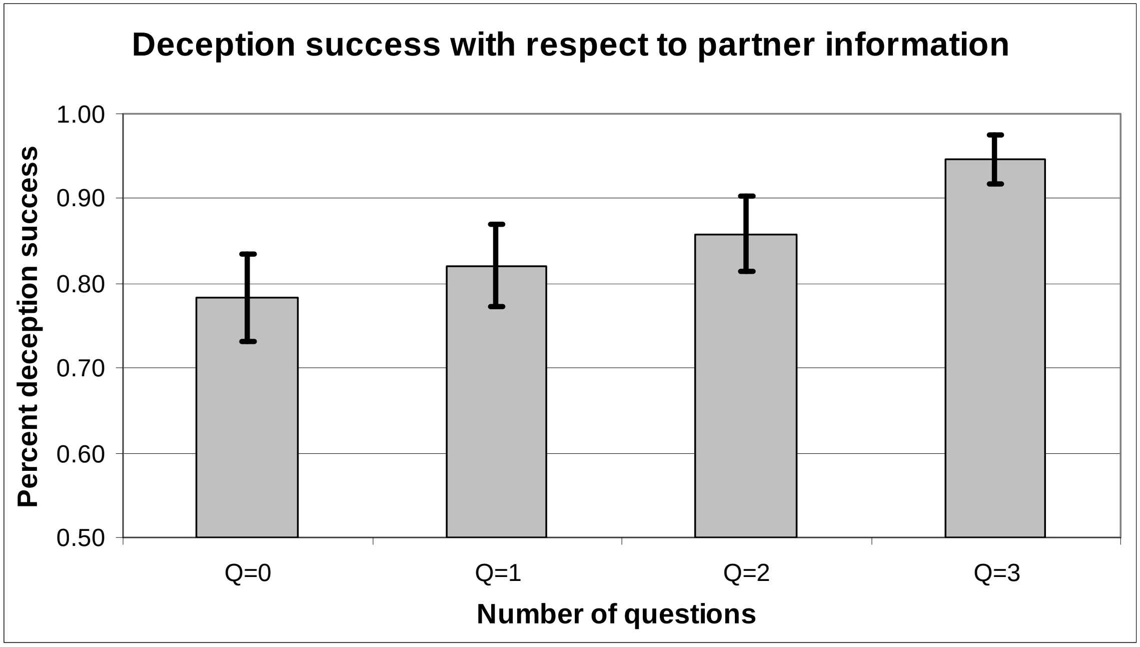


Figure 14 The graph above depicts the percent of successful deception for different numbers of asked questions. The confidence intervals indicate that a significant difference results when comparing zero asked questions to three asked questions. Asking three questions resulted in significantly greater deception success ($p < 0.01$ for all).

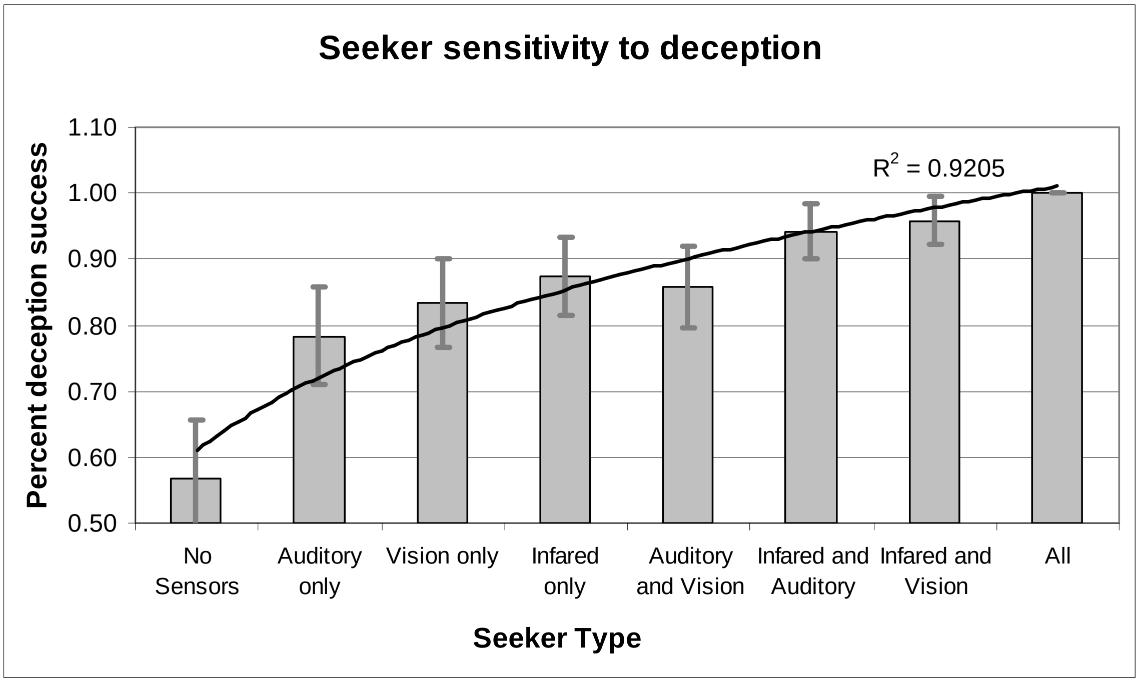


Figure 15 The graph above indicates the percent of successful deception for each different seeker type. When the seeker has no sensors the percent success is approximately 0.66, indicative of the unsuccessful deception. As sensors are added to the seeker, it becomes more susceptible to the deception. The trendline is a power function.

4.3.2 Robot Experiment

We also conducted a robot experiment to explore the impact of sensor and actuation noise on the learning and use of deception signals. In this experiment, the deceiver learns a set of three deception signals. The robot then attempts to use the learned signals to deceive the seeker.

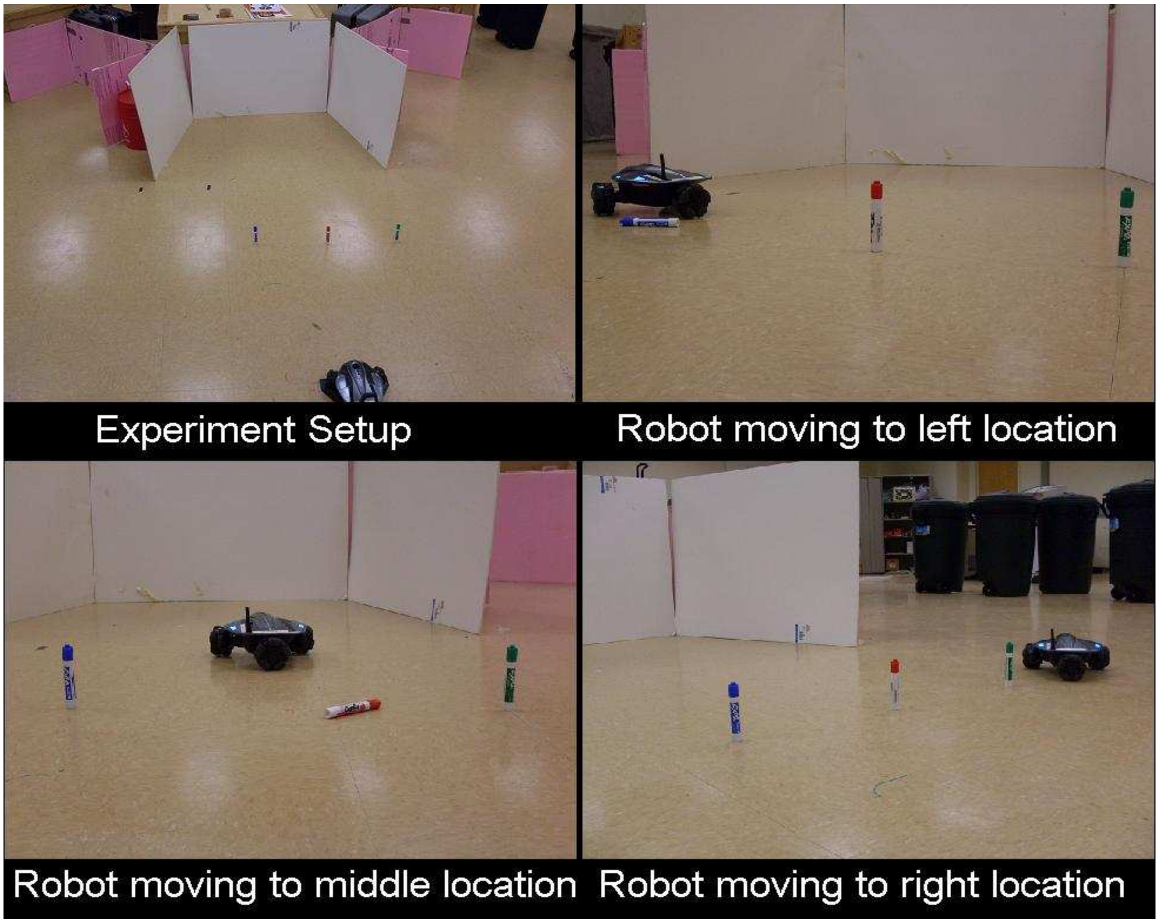

Experiment Setup
Robot moving to left location
Robot moving to middle location
Robot moving to right location

Figure 16 The images above depict the experimental setup for the robot experiment. The top left image shows the hider, the left, center, and right hiding location. The top right photo depicts the hider moving to the left hiding location. The hider knocks over the blue marker enroute to the location. The bottom left photo depicts the hider going to the center hiding location. In this case the hider knocks over the red marker. Finally, the bottom right photo depicts the hider moving to the right hiding location. In this case, the hider does not knock over any markers.

For this experiment, communications related to the hider’s position were accomplished by knocking over markers en route to the hiding location. As depicted in Figure 16 colored markers were positioned along the pathway to each hiding location. The robot, autonomously traveling to one of the locations, typically knocked over one or more of the markers. The presence or absence of standing markers thus indicated the hider’s location to a potential seeker.

Table 4 The table below lists the conditional probabilities learned by the hider and the seeker.

Hider and Seeker’s Learned Marker Conditional Probabilities						
X	$b = true$	$b = false$	$r = true$	$r = false$	$g = true$	$g = false$
$p_H(x = l)$	0.8	0.2	0.5	0.5	0.0	1.0
$p_H(x = c)$	0.0	1.0	1.0	0.0	0.0	1.0
$p_H(x = r)$	0.1	0.9	0.1	0.9	0.2	0.8
$p_S(x = l)$	0.7	0.3	0.1	0.9	0.0	1.0
$p_S(x = c)$	0.0	1.0	0.5	0.5	0.1	0.9
$p_S(x = r)$	0.0	1.0	0.0	1.0	0.2	0.8

During a learning phase, the hider first went to each location 10 times and recorded whether or not each of the three markers remained standing. The results are presented as conditional probabilities in Table 4. The conditional probabilities from Table 4 were used in conjunction with the graphical model in Figure 17. The junction tree algorithm was again used to perform inference on the model. For example, witnessing the pattern $\langle B = false, R = true, G = false \rangle$ indicates that the red marker was knocked down but the blue and green are standing and occurs when the hider traveled to the center location. Next, the junction tree algorithm is used to determined the marginalized probabilities of X: $p(X = l | B = f, R = t, G = f) = 0.06$, $p(X = c | B = f, R = t, G = f) = 0.94$, and $p(X = r | B = f, R = t, G = f) = 0$. Finally, given the conditional probabilities, the hider expects that a signal of $\langle B = false, R = true, G = false \rangle$ indicates the hider’s location to be center (the marginal probability of maximum likelihood). Hence, the signal $\langle B = false, R = true, G = false \rangle$ is a good false communication when the hider’s location is either left or right.

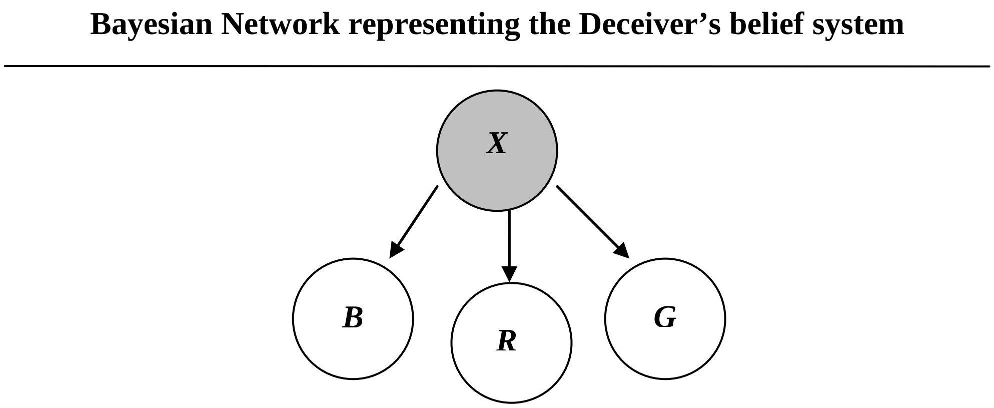

Bayesian Network representing the Deceiver's belief system

Figure 17 A Bayesian network representing the relationship of the random variables for the deceiver's belief system in the laboratory robot experiment. The variable $X = \{l, r, c\}$ again denotes the corridor the mark will choose to search in, $B = \{true, false\}$ denotes the blue marker, $R = \{true, false\}$ denotes the red marker, and $G = \{true, false\}$ denotes green marker.

Producing the false communications is similarly straightforward. The hider simply moves as if they are going to a location and then, once it reaches a point past the markers, changes course to some other location. Continuing the example from the prior paragraph, the hider moves as if hiding in the center location, typically knocking only the red marker down, then moving to either the left or the right hiding position ideally traveling along a path that avoids any markers. The hider's set of false communications was defined by selecting a pattern of knocked over markers that indicated a false hiding position for each possible hiding position.

The following procedure was used:

Hide and seek experimental procedure

1) The hider moves to each location 10 times recording which markers are standing afterward.
2) The seeker moves to each location 10 times recording which markers are standing afterward.
3) The hider generates a false communication for each location based on the results of step 1).
4) The hider creates an initial partner model representing the seeker with action set $A^M = \{\texttt{GoToAreaLeft, GoToAreaCenter, GoToAreaRight}\}$ and the utilities values depicted in Figure 13.

5) The hider randomly selects an action from the set $A^D = \{$`HideInAreaLeft, HideInAreaCenter, HideInAreaRight`$\}$

6) The hider creates the true matrix with respect to its action selection from step 6). Figure 3 depicts the result for the action `HideInAreaLeft`.

7) The hider uses the partner model created in step 5) and the true matrix from 6) with constants $k_1 = 20, k_2 = 20$ as input to the algorithm from Figure 8.

8) The algorithm results in the selection of a false communication from $\Gamma = \{y_0, \ldots, y_M\}$. The hider produces the false signal by initially moving to the location and knocking over markers.

9) The hider performs the action $A^D = \{$`HideInAreaLeft, HideInAreaCenter, HideInAreaRight`$\}$.

10) The seeker perceives which markers have been knocked over (the hider's false communication signal).

11) The seeker selects a corridor and searches the corridor.

12) If the seeker selects the correct corridor, the hider's deception is recorded as a failure. Otherwise, the hider's deception is recorded as a success.

During a similar but separate learning phase the seeker moved to each of the locations 10 times and recorded whether or not each of the three markers remained standing. The results of the seeker's learning are presented in Table 4 (as p_S). The results from Table 4 were used in conjunction with the seeker's version of the graphical model depicted in Figure 17 representing the seeker's belief system of the hider's location. The junction tree algorithm was then used by the seeker to calculate the marginal probability of X given evidence in the form of standing markers. The seeker then searched in the area with maximal marginal probability. Color blob detection was used to produce these evidence vectors. A procedure for the hider and the seeker is presented above.

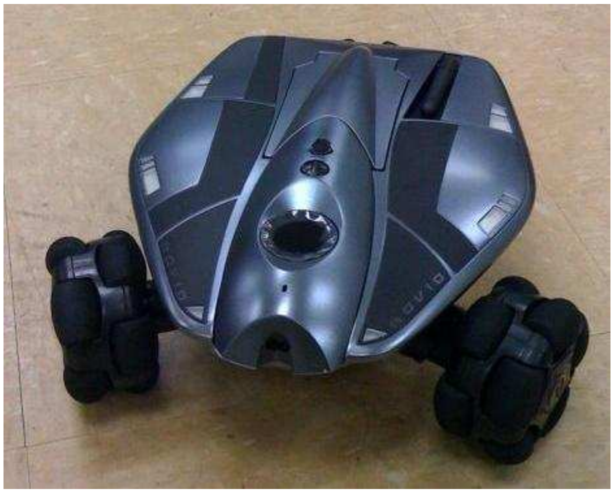


Figure 18 WowWee’s Rovio robot.

WowWee’s Rovio robot was used in these experiments for both the role of the hider and the seeker (Figure 18). This robot is a wheeled robot with mobility over smooth flat surfaces. The robot’s camera is a 640 by 480 pixel webcam mounted on the top of the robot’s extendable neck. Communication with the robot is accomplished via the robot’s wireless network card. The Rovio comes with a docking station and infrared beacon easing the robot’s navigation task back to the docking station.

We ran twenty trials using the hide and seek experimental procedure. The hider randomly selected a hiding location, produced the false communication, and then moved to the hiding location. The seeker recognized the signal, produced the evidence vector, determined the best location to search, and finally moved to that location. We found that the deception worked in 75 percent of the trials (15 of 20) and failed in 5 of the 20 trials. Each of these 5 failures resulted from the hider’s inability to knock over the correct markers to produce the desired deceptive communication. In two of these cases the hider attempted to signal that it was in the center location by knocking over the red marker, but missed knocking over any markers. In the remaining three cases, the hider accidentally knocked over a marker which signaled its location. Overall, the hider produced the incorrect signal in 7 of the 20 trials. Two of the incorrect signals, simply because of chance, did not impact the deception.

Overall, the experiment demonstrated the learning and use of deception signals in a noisy environment on a real robot. Moreover, the results are a preliminary indication that the techniques and algorithms described in this paper can be fruitfully used to produce deceptive behavior in a robot.

6. The Ethical Implications of Deceptive Robots

One might question the intent behind creating deceptive robots in the first place. While obviously there is utility in military situations, as deception has been used to advantage there throughout recorded history, it is entirely possible that the tools and techniques used to understand both when a robot should deceive and the methods to accomplish such deception could conceivably be used for nefarious purposes.

Our laboratory has spent considerable effort in exploring and understanding many of the ethical quandaries associated with the effects of robots and society, e.g., (Arkin 2009, Arkin 2008). We assume that techniques for deception can and will be further developed in the future and this research serves as a stake in the ground, indicating the possibility of creating such a potentially unethical capability in robotic systems. As a result, we strongly encourage discussion about the appropriateness of this and other related areas of robot ethics by the appropriate communities (e.g., Euron 2007) and relevant professional societies, to determine what, if any, regulations or guidelines should constrain the designers of these systems. It is crucial that these considerations be done proactively rather than reactively in order to ensure that these creations are consistent with the overall expectations and well-being of society.

7. Summary and Conclusions

This article arguably represents the first detailed examination of robot deception. Our exploration of the topic began with a working definition borrowed from biology. We used this definition and the interdependence theory framework presented in [6] to reason about, develop, and test algorithms which we believe will allow a robot to recognize when a situation warrants the use of deception and how a deceiver can and should select a false communication. Our results show that:

1) a situation's location in interdependence space can be used to determine if a robot or agent should act deceptively;
2) a deceiver's knowledge about the mark can aid in determining which false communication the deceiver should use; and
3) learned communications can be used as deceptive signals by a robot.

We have also discussed some of the ethical implications related to the creation of robots capable of deception.

The algorithms presented herein assume that outcome matrices representing the interactions faced by the robot can be created. Previous work serves as evidence that outcome matrices reflecting these situations can indeed be created [1]. Our algorithm for acting deceptively also assumes that the deceiver has a model of the mark. Our results have shown that the information within the deceiver's model of the mark is an important factor in determining the deception's success or failure. We did not explore the implication of partner modeling on the part of the mark.

Our experiments were developed to examine the algorithms and their related underpinnings. As such, they do not represent the final word on robots and deception. It is our hope that other researchers will continue to explore this topic. We are currently working on a software release to promote such work. With respect to our algorithm for determining if a situation warrants deception, we feel that psychologically grounded experimentation is necessary to determine the correlation between situations our algorithm selects as warranting deception and a random human subject population. Moreover, while our results linking the success of deception to one's knowledge of the mark may appear trivial, these results are critical in that they conceptually and empirically link theory mind to the use and success of deception.

This research highlights and reinforces the role that theory of mind plays in deception. The acting deceptively algorithm was developed around the notion that the deceiver uses a model of the mark to decide how to deceive. Moreover, we have intentionally used a broad definition of deception in the hope of applying our results as generally as possible. While some of the mechanisms and representations, such as the structure of the Bayesian network, used in the experiments were tied to a particular problem, for the most part, this work stands as a generally applicable computational foundation for understanding the phenomena of deception.

Research exploring the use of deception by robots is potentially important for several different application areas. Military applications are an obvious possibility. Less obvious applications could possibly aid a robot's management as situations within assistive or search and rescue. A search and rescue robot may need to deceive in order to calm or receive cooperation from a panicking victim. Socially assistive robots are expected to provide patients in a healthcare setting with personalized care. Generally, one would not expect the goals of a robot trying to help to be in conflict with a patient. But there are cases in which this does happen. Again, patients suffering from acute trauma may need to be deceived in order to receive cooperation. Overall, for many social robotics and multi-robotics application areas the use of deception by a robot may be rarely used, but nonetheless an important tool in the robot's interactive arsenal, just as it has been with intelligent systems throughout the animal kingdom.

Acknowledgments

This work is dedicated to the memory of Caryl Rusbult (1953-2010) whose research on the theoretical understanding of relationships inspires our continued investigation of an interdependence framework for artificial social intelligence. The authors would like to thanks Zsolt Kira and Patrick Ulam for their helpful suggestions and continued words of inspiration to complete this work. This work was supported in part by Office of Naval Research under MURI Grant # N00014-08-1-0696.

References

1. Wagner, A.R., *Creating and Using Matrix Representations of Social Interaction*, in *Proceedings of the 4th International Conference on Human-Robot Interaction (HRI 2009)*. 2009: San Diego, CA. USA.
2. Cheney, D.L. and R.M. Seyfarth, *Baboon Metaphysics: The Evolution of a Social Mind*. 2008, Chicago: University Of Chicago Press.
3. Wagner, A.R. and R.C. Arkin, *Analyzing Social Situations for Human-Robot Interaction.* Interaction Studies, 2008. **10**(2): p. 277–300.
4. Hauser, M.D., *Costs of deception: Cheaters are punished in rhesus monkeys (Macaca mulatta).* Proceedings of the National Academy of Sciences, 1992. **89**: p. 12137-12139.
5. Bond, C.F. and M. Robinson, *The evolution of deception.* Journal of Nonverbal Behavior, 1988. **12**(4): p. 295-307.
6. Wagner, A.R., *The Role of Trust and Relationships in Human-Robot Social Interaction*, in *School of Interactive Computing*. 2009, Georgia Institute of Technology: Atlanta, Georgia.
7. Gerwehr, S. and R.W. Glenn, *The art of darkness: deception and urban operations*. 2000, Santa Monica, CA: Rand Corporation.
8. McCleskey, E., *Applying Deception to Special Operations Direct Action Missions*. 1991, Washington, D.C. Defense Intelligence College.
9. Whaley, B., *Towards a General Theory of Deception.* The Journal of Strategic Studies, 1982. **5**(1): p. 178-192.

10. Ettinger, D. and P. Jehiel, *Towards a theory of deception*. 2009, ELSE Working Papers (181). ESRC Centre for Economic Learning and Social Evolution, London, UK.

11. Osborne, M.J. and A. Rubinstein, *A Course in Game Theory*. 1994, Cambridge, MA: MIT Press.

12. Spence, M., *Job Market Signaling.* Quarterly Journal of Economics, 1973. **87**(3): p. 355-374.

13. Floreano, D., et al., *Evolutionary Conditions for the Emergence of Communication in Robots.* Current Biology, 2007. **17**(6): p. 514-519.

14. Kelley, H.H. and J.W. Thibaut, *Interpersonal Relations: A Theory of Interdependence*. 1978, New York, NY: John Wiley & Sons. 341.

15. Gibbons, R., *Game Theory for Applied Economists*. 1992, Princeton, NJ: Princeton University Press.

16. Sears, D.O., L.A. Peplau, and S.E. Taylor, *Social Psychology*. 1991, Englewood Cliffs, New Jersey: Prentice Hall. 634.

17. Axelrod, R., *The Evolution of Cooperation*. 1984, New York: Basic Books.

18. Rusbult, C.E. and P.A.M. Van Lange, *Interdependence, Interaction and Relationship.* Annual Review of Psychology, 2003. **54**: p. 351-375.

19. Kelley, H.H., et al., *An Atlas of Interpersonal Situations*. 2003, New York, NY: Cambridge University Press. 506.

20. Powers, A. and S. Kiesler, *The advisor robot: tracing people's mental model from a robot's physical attributes*, in *1st ACM SIGCHI/SIGART conference on Human-robot interaction*. 2006: Salt Lake City, UT, USA.

21. Norman, D., *Some Observations on Mental Models*, in *Mental Models*, D. Gentner and A. Stevens, Editors. 1983, Lawrence Erlbaum Associates: Hillsdale, NJ.

22. Akerlof, G.A., *The Market for 'Lemons'. Quality Uncertainty and the Market Mechanism.* Quarterly Journal of Economics, 1970. **84**(3): p. 488-500.

23. Wagner, A.R. and R.C. Arkin. *Robot Deception: Recognizing when a Robot Should Deceive*. in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*. 2009. Daejeon, Korea.

24. Yamagishi, T., *Trust as a Form of Social Intelligence*, in *Trust in Society*, K.S. Cook, Editor. 2001, Russell Sage Foundation: New York, NY. p. 121-147.

25. Stoytchev, A. *Behavior-Grounded Representation of Tool Affordances*. in *IEEE International Conference on Robotics and Automation (ICRA)*. 2005. Barcelona, Spain.

26. Jordan, M., et al., *An introduction to variational methods for graphical models*. Learning in Graphical Models, ed. M. Jordan. 1998, Cambridge: MIT Press.

27. Arkin, R.C., *Behavior-Based Robotics*. 2 ed. Intelligent Robots and Autonomous Agents, ed. R.C. Arkin. 1999, Cambridge, MA: The MIT Press. 491.

28. MacKenzie, D.C., R.C. Arkin, and J.M. Cameron, *Multiagent Mission Specification and Execution.* Autonomous Robotics, 1997. **4**: p. 29-52.

29. Trafton, J.G., et al. *Children and robots learning to play hide and seek*. in *1st ACM conference on Human-robot interaction*. 2006. Salt Lake City: ACM Press.

Biographies

Alan Wagner is a post-doctoral researcher under the supervision of Dr. Lora Weiss at the Georgia Institute of Technology Research Institute. He recently earned his doctoral degree from Georgia Institute of Technology working under the supervision of Dr. Ronald Arkin. His doctoral research explored the development of a theoretical framework for human-robot interaction based on related work in social psychology and game theory. His current research investigates the development of computational methods that will a robot to recognize and reason about social situations involving trust, deception, and stereotypes. He is a member of the IEEE, ACM, and AAAI.

Ronald C. Arkin is in the College of Computing at the Georgia Institute of Technology where he holds the rank of Regents' Professor and is the Director of the Mobile Robot Laboratory. He also serves as the

Associate Dean for Research in the College of Computing at Georgia Tech. During 1997-98, Professor Arkin served as STINT visiting Professor at the Centre for Autonomous Systems at the Royal Institute of Technology (KTH) in Stockholm, Sweden. From June-September 2005, Prof. Arkin held a Sabbatical Chair at the Sony Intelligence Dynamics Laboratory in Tokyo, Japan and then served as a member of the Robotics and Artificial Intelligence Group at LAAS/CNRS in Toulouse, France from October 2005-August 2006. Dr. Arkin's research interests include behavior-based reactive control and action-oriented perception for mobile robots and unmanned aerial vehicles, hybrid deliberative/reactive software architectures, robot survivability, multiagent robotic systems, biorobotics, human-robot interaction, robot ethics, and learning in autonomous systems. He has over 170 technical publications in these areas. Prof. Arkin has written a textbook entitled Behavior-Based Robotics published by MIT Press in May 1998, co-edited (with G. Bekey) and a recent book published entitled Governing Lethal Behavior in Autonomous Robots published by Chapman-Hall (Taylor & Francis). Dr. Arkin serves/served as an Associate Editor numerous journals and is the Series Editor for the MIT Press book series Intelligent Robotics and Autonomous Agents. Prof. Arkin serves on the Board of Governors of the IEEE Society on Social Implications of Technology and served two terms on the Administrative Committee of the IEEE Robotics and Automation Society. He was elected a Fellow of the IEEE in 2003, and is a member of AAAI and ACM.