A General Architecture for Robotics Systems: A Perception-Based Approach to Artificial Life

Rupert Young*
Perceptual Robots

Keywords
Robotics, perceptual control theory, behavior-based robotics, embodiment, situatedness, controlled variables

Abstract Departing from the conventional view of the reasons for the behavior of living systems, this research presents a radical and unique view of that behavior, as the observed side effects of a hierarchical set of simple, continuous, and dynamic negative feedback control systems, by way of an experimental model implemented on a real-world autonomous robotic rover. Rather than generating specific output from input, the systems control their *perceptual inputs* by *varying* output. The variables controlled do not exist in the environment, but are entirely internal perceptions constructed as a result of the layout and connections of the neural architecture. As the underlying processes are independent of the domain, the architecture is universal and thus has significant implications not only for understanding natural living systems, but also for the development of robotics systems. The central process of *perceptual control* has the potential to unify the behavioral sciences and is proposed as the missing behavioral principle of Artificial Life.

1 Introduction

The theory and research described in this article present a view of behavior as being a side effect of a hierarchical set of negative feedback systems that control, and maintain, their perceptions of the world at internally specified values. The difference between the current and goal values of the perception generates an output, which in turn, by way of the environment, affects the perception, bringing it closer to the goal. This is the basic form of a perception controller within which there is no starting point, but all elements operate simultaneously within a continuous loop of causation. The hierarchical arrangement of the controllers, where the higher-level perceptual inputs are derived from the combination of perceptions from lower levels, results in the capability to control an increasingly rich and abstract set of high-order perceptions.

The underlying foundation, *perceptual control theory* (PCT) [40], describes how the purposeful behavior of living systems could arise from such a configuration of perception controllers, in contrast [33] to conventional theories of the general linear model [27], neurobiological control [16], the planning of actions [35], prediction [20], synthetic psychology [8], and behavior-based robotics [10], which follow a stimulus-response model whereby output is generated as a function of input.

* Perceptual Robots, Windsor SL4 1TB, United Kingdom. E-mail: rupert@perceptualrobots.com

© 2017 Massachusetts Institute of Technology doi:10.1162/ARTL_a_00229

The main experiment presents an autonomous robotic rover that exhibits behaviors of target location and pursuit, obstacle avoidance, and recovery from deadlock. However, these have not been explicitly encoded within the system, but are a consequence, from the observer's point of view, of the interaction of a set of perception controllers within a dynamic and unpredictable environment.

The successful implementation of the principles of PCT in a real-world robot provides support for the biological hypothesis that perceptual control is the central process of behavioral systems. Specifically, the support is provided by demonstrating a number of characteristics of the architecture:

- that a complex behavioral system can be composed entirely of a set of simple perception controllers,
- that perception controllers are adaptive and compensate for unknown disturbances, which therefore do not need to be modeled,
- that high-level perceptions, which are functions of lower-level perceptions, can be controlled—in this case *conditional*, *sequence*, and *internal state* perceptions, which are crucial for complex behavior and for the scalability of the approach—and
- that a perceptual control system is truly autonomous, as goals are embodied within the system rather than in the mind of the programmer.

Overall, this piece of work is the groundwork in a new line of research, and its purpose is twofold: to verify the underlying theory and to define a generalized architecture for future development. The implemented system provides some initial validity of PCT by supporting the main hypothesis that observed behavior is the result of a collection of simple feedback processes that control their perceptions. The methodology as a whole addresses some of the key challenges of *artificial life* (ALife), and perceptual control is proposed as providing the missing [9] principle that differentiates the behavior of living from nonliving systems. The issues raised suggest that conventional views of intelligence and behavior are fundamentally flawed and need to be reevaluated. With regard to robotic systems, the current architecture provides a blueprint for building artificial systems, as a precursor to implementing more sophisticated systems that include memory and learning.

Before proceeding further in this article, the reader is invited to view the video footage [59] of the main experimental demonstration, in order to form their own opinion of what the system is doing and how it is doing it, for later comparison with the actual methodology.

A couple of language clarifications regarding the usage of the words *behavior* and *control* are in order at this point. Both can be used in a general sense and in a more specific sense. *Behavior* can generally mean everything that humans, or animals, do, or it can mean the specific output or actions made at a particular point. Both senses are employed in common usage, and this article is no different, but the context should make it clear which sense is intended. *Control* can mean to generally manipulate something, or someone, as in controlling an airplane. It can also be used in a specific technical sense within the context of a *control system*, whereby a particular variable is being controlled. In this article only the technical sense will be used.

The next three sections provide an overview of the methodology and its novelty, highlighting the power and breadth, yet simplicity, of its application to general behavior, and contrasting it with conventional approaches. Section 5 summarizes that discussion with a list of key attributes and benefits of the methodology in the context of its consideration as an artificial life framework.

Sections 6–8 describe the main experimental application of the methodology and architecture to a real-world robot rover. An additional experiment, in Section 9, gives some insight into the nature of perceptions at levels higher than those at the interface with the environment, indicating that perception and perceptual control are not restricted to primitive sensory information, but that the neural architecture can construct and control perceptions that have no direct correlate in the external environment. Perceptions higher up the hierarchy are dependent upon the lower perceptual infrastructure, resulting in an architecture that is intimately coupled with the physical environment, but is also, theoretically, able to encompass abstractions necessary for higher-level cognition and intelligence.

Section 10 discusses the results of the main experiment, emphasizing that observed behavior does not reflect the underlying, internal intentions and system architecture. The concluding section, 11, suggests the directions in which the methodology could be taken forward and proposes that it has profound implications relevant to our understanding of behavioral and intelligent systems and therefore to how they should be artificially constructed. Finally, an appendix details the configuration of the control systems used in the main experiment.

2 Perceptual Control Theory Overview

2.1 Control of Perceptual Goals

Even a cursory glance at an everyday example of behavior demonstrates that the stimulus-response view does not provide an adequate explanation. Consider a man in a bar who switches on the television set in order to watch football. Some of the people in the bar turn to look at the screen, which seems to be a classic response to the stimulus of the football on the television. But what about those who did not turn to watch—the same "stimulus," but no response? And how about the guy who switched on the television in the first place—no stimulus, but a "response"?

PCT, as a theory of purpose-driven behavior, can easily dispel any confusion. All of the participants have purposes, internal motivations, which are individual to themselves. The first person has a goal to watch football and so acted to achieve that goal by switching on the television. The other watchers have an ongoing football-watching goal, which came into focus when the football became available. The remaining people lacked any goal to watch football and so showed no interest.

In contrast to the conventional view of behavioral output being the goal, and a function of the input, the operation of a perceptual control system is not to produce or regulate a correct, or specific, *output*, but to vary the output to achieve the desired *input*. The iris system that controls the light falling on the retina does so by varying the size of the pupil. The system has physical limitations, but there are other ways humans can control that light: squinting, wearing a baseball cap, switching a light on, shading the eyes with a hand, wearing sunglasses, or turning a dimmer switch. The *means* vary, but constant throughout is the common goal of the amount of light falling on the retina.

Living systems are purposeful agents and act in the world in order to fulfill those goals. Whether it be at a low level of controlling the sound of a musical note or at a high level of the meaning inferred from language, perception is the only access we have to what is happening in the world. Therefore, goals must be perceptual in nature, as perception is the only way of determining whether or not we have achieved those goals.

2.2 The Control System

The main components of a generalized PCT control unit are shown in Figure 1. The lines represent signals and the circles functions, which transform one or more signals into other signals. The components approximate to neurons of the nervous system and the type of basic operations thought to be performed by them, with the signals being the equivalent of neural impulses.

Within the triangle are:

- the perceptual function, which emits a perception signal, *p*, based upon its input signals, perhaps a weighted sum,
- the reference signal, *r*, which represents the desired value of the perceptual signal,
- the comparator (comparison function), which compares (usually by subtraction) the reference and perceptual signals, and emits a corresponding error signal, *e*, the magnitude of which represents how far the perception is from the desired (reference) value, and

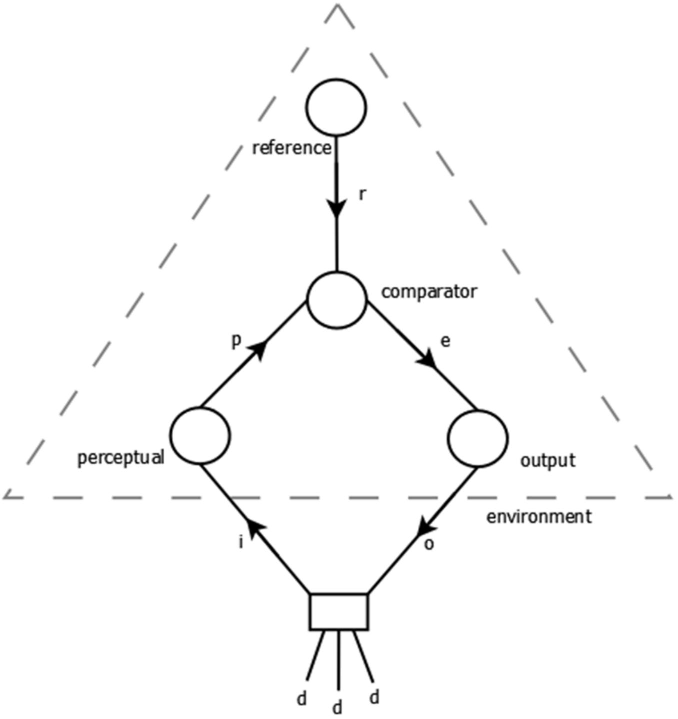

Figure 1. The basic perceptual control unit. The area within the triangle is a subelement replicated throughout the hierarchy within the organism; see Figure 3. The area below the triangle is the feedback path through the environment. Each function (the circles) transforms one or more signals (the lines) to other signals. The single-letter notation refers to the signals output from the corresponding functions: *p* = perception, *r* = reference, *e* = error, and *o* = output. *i* is the input to the control system, which is a combination of the effects of the output of the system and any environmental disturbances, which are denoted by *d*.

- the output function, which transforms the error signal into some output, *o*, of the whole unit. At the interface with the physical environment this might be an indication to change muscle tension.

The operation of any PCT system requires *feedback*: that the output affect the perception, in such a way that it is brought closer to the reference state; thus the error is reduced (the feedback is negative). It is not necessary for the system to know in advance what action is required to achieve its goal. The choice of action is an inherent part of the negative feedback system, which keeps the error to a minimum. If action taken is in a direction that increases the error, then there will be increased effort (output from the system) in the direction that would bring the error back down to zero. For example, when acting to pick up a mug of tea, if your hand were to move away from the mug, the control system error would increase, which would increase the output that affects, by feedback, the perception of your hand in relation to the mug. As the feedback is negative, the direction of the effects of the output is such that the error is reduced and the perception (of the hand-mug separation) is brought closer to its goal.

2.3 A Simple Definition

It should be noted that the functions and signals in the control systems do not act in a serial manner, but all act simultaneously and continuously. In fact, the basic control unit can be mathematically described by two simple simultaneous equations,

$$i = o + d \tag{1}$$

and

$$o = g \cdot e \tag{2}$$

where $e = r - p$, $p = f(i)$, and g is the gain, or sensitivity, of the system.

Equation 2 gives the proportional form of the equation, which also commonly takes a leaky integration form,

$$o = o + (g \cdot e - o)/l, \tag{3}$$

where l is the leak rate.

As Equation 3 *adds* a proportion of the error to the output, this can result in zero error, but nonzero output.

2.4 Continuous Adaptation

As can be seen from Figure 1 and Equation 1, the perception is affected not only by the output of the control system, but also by any environmental disturbances, that is, variables that are not directly sensed. For example, when driving, a goal is to control the perception of the position of the car on the road. However, there are many factors, such as wind, road conditions, and tire pressures, that may have an effect on the position. These disturbances are only indirectly detectable, through their effects on the controlled variable, the car's position. However, there is no need to detect disturbances, as the outcome of the operation of a perceptual control system is to *protect* the controlled variable from any disturbances. If anything were to disturb the variable, then (within physical limitations) the error would increase, and so there would be compensatory output to bring the perception back to its desired state. It follows from this that a control system can be in a state of rest, in that there is no error, but there could be nonzero output, which would reflect the sum of disturbances. For example, the steering wheel may be turned to counteract the wind acting upon the side of the car.

A lack of (observed) behavior does not indicate lack of control. A man standing at a bus stop may appear to be doing nothing, but he is actually controlling many things simultaneously, such as his sense of balance and his upright body position, which would wilt and wither to the ground if tension were not applied to muscles in order to maintain that perceived position by counteracting the disturbance of the effects of gravity.

2.5 A Hierarchy of Control

The real power of PCT derives from the replication of the basic control unit at many different levels in a neural hierarchy, according to *hierarchical perceptual control theory* (HPCT), enabling control of high-order perceptions resulting in the complex behavior we observe in living systems.

At the lowest level there is the interface with the external environment, as shown in Figure 1 at the base of the triangle, where the inputs are the signals perceived by the sensory system and the outputs affect the musculatory system. The box represents the feedback path—that is, everything that physically affects the perception in the environment, which includes the output and any disturbances. For higher-level control units the feedback path comprises all corresponding lower-level units as well as the external environment.

For any unit in the hierarchy the output could be thought of as setting, or contributing to, a lower-level goal (reference) that would help the unit achieve its own goal. To some degree each unit can be regarded as an independent module that only need be concerned with varying its output in order to keep its error low, with the perception matching the reference. The reference itself may be changing dynamically according to the varying output from above, meaning that the control system is in a constant state of flux.

However, the overall organization of the hierarchical system needs to be such that the associated lower control systems contribute to achieving the goal of the higher system. The development of a specific organization is driven by the overall error within the system, which changes the structure so that successful control is achieved (the error is kept low). As with an individual system, this is achieved without the requirement for external, or a priori, knowledge of what the changes or action need to be. How the self-organizing developmental process might work is discussed in more detail in Section 2.8, on learning and memory.

Perceptions at the different levels in the hierarchy can be viewed as different *types* [43], becoming more abstract the higher we go. At a low level we might control our perception of an object by manipulating it until we see it from the desired perspective. We would, for example, turn a shoe around in our hands until we perceive it in the right way prior to putting it on. An object is a *configuration* perception, composed of lower-level *sensation* perceptions. At a high level we may have a perception of ourself as a "good" person, for which we would have a set of lower-level *principle* perceptions, which we also control, such as honesty or "thou shalt not kill." In the middle we might be controlling a perception of the recipe for making a pizza, which is the control of a *perception* of a *program* consisting of lower-level perceptual goals of *conditions* and *sequences*.

Crucially, in contrast to conventional wisdom, what is specified in a control hierarchy from a higher level is not an action or output, but a *goal*: a value of the reference signal defining what the system should *perceive*.

2.6 Multi-layered Simplicity

HPCT can be regarded as a one-dimensional, multi-layered approach. One-dimensional perceptual control systems are simple both in concept and in implementation; each only need act to reduce its own error by controlling its individual perceptions. If there is any non-orthogonal interference (conflict) between the systems, this may be resolved by being perceived and controlled by a higher-level system, which itself is one-dimensional.

There are some similarities between robotic visual servoing [14] and basic PCT principles of error reduction. Typically, though, visual servoing, whether based upon features in images (image-based visual servoing) or upon the estimation of 3D parameters (position-based visual servoing), uses single-layer, multidimensional transformations between models of a robot, a camera, and the world [36], leading to more complex processing. The PCT approach is not limited to visual servoing. The general robotic architecture defined here, and derived from PCT, applies in principle to all forms of behavior.

2.7 A Universal Process

Perceptual control is not an isolated process, but is common to all types and levels of behavior and cognition, in humans and other animals, providing a biologically, sociologically, and psychologically plausible model [13, 54] of the architecture and function of the nervous system.

Many examples of negative feedback control behavior can be found throughout the animal kingdom, such as the visual orientation towards prey of the praying mantis [55], play fighting of rats [39], birds in a flock controlling their proximity to each other [46], humans doing the same in a crowd [44, Chapter 9], or talking face to face or catching a baseball [49]. However, PCT proposes that *all* behavior is governed by negative feedback control systems, even though not immediately apparent; it is just a matter of determining the variable under control. The concept of feedback control is well recognized in the field of neurobiology, but reference goals are often misrepresented as being external to the organism [41], and the role of control is misconstrued as being to regulate output [16].

Once you don your PCT glasses [30] and look at the things that living systems do, the control of perceptions can be seen everywhere: control of perceived thirst by controlling the sight of a bottle of water being raised to the lips and the pressure of the fingers on the plastic; control of the sequence of letters that appear on the computer monitor through control of the perceived position of the fingers on the keyboard, in order to control and convey the desired meaning of words and sentences that adequately describe perceptual control systems as a means to control the desire to understand living systems and inform others; control of the feeling of heat on the body by selecting an Internet café with air conditioning rather than just a fan; control of the feeling of hunger while working, by having just eaten; control of a desire to travel while working by writing a research article while cycling around India, which also controls the desire for a lifestyle based on health and exercise; control of a perception of honesty by correctly reporting the number of sheets of paper printed; control of a principle of equal rights by ongoing support for human rights groups; control of political and social system concepts by voting for groups, and reading newspapers, with corresponding ideologies. Multiple perceptions at many different levels of abstraction all being controlled at the same moments by a single (human) organism, some lasting a few seconds, others a lifetime. When scrutinized, all behavior can be seen in the light of a multitude of purposes and goals, which equate to perceptions being controlled.

2.8 Learning and Memory

Learning within PCT is a process of reorganization [40] of the structure of the control system hierarchy, with a central mechanism being the error-driven change in weights between neural connections in a manner akin to that employed within standard neural networks [37]. However, rather than the process converging on a mapping between between input and output patterns, reorganization continues until behavior appears that reduces the error to a minimum. A structure that produces behavior that reduces error persists, whereas one that does not is reorganized out of existence.

In the case of PCT the error in question is intrinsic to the state of the organism, which may affect the physiology of the organism or may be an ongoing, persistent error within a subset of control systems. The change mechanism itself may be asymmetrically random [44, Chapter 7] in that the frequency of change is lower if the error is reducing.

An arm control simulation with 14 degrees of freedom [44, Chapter 8] demonstrates the reorganization process. The output weights change automatically and gradually, based upon the overall system error, and converge on a structure that results in good quality control.

PCT further suggests a role for consciousness and volition within learning, as a way of enhancing, or directing, a purely random process. Through conscious attention the system is able to focus on a specific area, change behavior at will, and immediately perceive the effects of the changes. This is, in effect, a quick and nonrandom way of selecting lower-level perceptual goals that lead to higher-level success. For example, when having in mind the sound that should be perceived when playing a guitar chord, attention can be focused on the perception of the individual finger positions and on consciously adjusting them until the played chord sounds as desired.

2.9 Evolutionary Context

PCT also fits well in the context of evolution by natural selection, which can be seen as the development of organisms with the neural architecture that has the abilities (or the potential to learn the abilities) to control relevant perceived variables. In line with *foraging theory* [50], these variables can be seen as enabling the acquisition of energy in more efficient ways, which in turn allows them to live long enough to procreate. For example, an organism such as the *zooplankton* [22] acquires its energy by absorbing light, and moves around, seemingly, according to the light available. Conventionally this would be seen as locomotion derived from an external signal. PCT indicates a different way of looking at the details of the behavior involved. According to PCT it is the energy level (or equivalent) that is being controlled, and the organism moves in order to keep that (internal) variable high. As it is able to perceive the direction of light, it is conceivable that the zooplankton also controls its swimming direction in order to serve its higher-level goal of the control of energy. The conventional view presents organisms as purposeless systems merely reacting to environmental conditions without regard to the effects of the actions and reactions. With such a strategy it would seem purely accidental that the zooplankton would acquire energy. The PCT perspective, however, suggests that even the lowly zooplankton has purposes, albeit very simple ones, and is able to act in the world in order to control and achieve those goals.

The PCT view of living organisms as systems with explicit internal motivations, as creatures that are purposefully scurrying, or swimming, or flying, or driving around in the world to achieve desired goals, is a more credible, rich, and comprehensive explanation of behavior than the impression given by the stimulus-response perspective of animals as automatonlike systems blindly responding to environmental stimuli without any internal purpose.

3 A Brief History of Conventional Approaches

3.1 Traditional Artificial Intelligence

From the early days of the *artificial intelligence* (AI) discipline, in the decades following the middle of the 20th century, research was heavily influenced by the computational approach advocated by Alan Turing [52] and what could be achieved with the main tool available, the computer [11, 12]. The outcome was that the issues that were addressed were associated with what were thought of as intelligent, high-level cognitive abilities, such as problem solving, theorem proving, natural-language processing, and—a modern incarnation—argumentation [6].

The "problem" of AI was seen as one of knowledge representation and symbolic reasoning, the main technique being *search* through the possible states. If only, it was thought, the states of the world could be well represented by a set of symbols, they could be manipulated by search techniques and the relevant set of knowledge rules to generate a new, desired state, along with the path to that state, whether it be a chess-playing program or a robot manipulating a set of blocks within a world model [47]. However, realization grew that this type of top-down approach was mired in well-constrained "toy" worlds and reflected a paradigm akin to abstract formal logic, lacking semantics, more than to one grounded [19] in reality. Real worlds are messy, chaotic, and dynamic. Animals, and people, more often than not do not perform in the most optimal and logical manner [38, 51]. The sensory inputs available are myriad in number and, by themselves, primitive and meaningless in nature [45].

3.2 Behavior-Based Robotics

In the last two decades of the 20th century, something of a conceptual, and pragmatic, revolution took place in AI, exemplified by the work of Professor Rodney Brooks at MIT. He proposed *behavior-based robotics* (BBR), a bottom-up approach of building real-world systems with simple behavioral capabilities to interact with the real world [10],[1] incrementally adding layers of more complex abilities to expand the sophistication of the system. This ALife approach involved principles and characteristics such as situatedness and embodiment [1], distributed control, hierarchies, parallel processing, autonomy, emergence, and the use of the real world as its own model.

3.3 Model-Based Control

Although BBR represented a progression to real-world robots, the approach was found to be limited to simple behaviors, mainly on account of the primitive representation embodied within the systems. So, more recently, there has been something of a return to traditional AI, with the reinstatement of world models providing a richer representation for the purposes of defining the sensorimotor transformations between input and output [15, 24]. Advanced technology, including significantly more powerful processors, has led to some promising progress with this approach [3, 28, 15, 18].

A major problem, though, is the difficulty in forming accurate models of the physical structure and dynamics of the world. This has partially been addressed by incorporating feedback within the systems so that the world state estimation is updated, potentially correcting for inaccuracies. However,

[1] In the title of his article Brooks uses "control system" in the general sense.

there still remains the significant difficulty of the exponential rise in the complexity of the models with the increase in degrees of freedom [53, 2].

The general practicality of the model-based control approach is in question when specific models need to be manually devised for each solution [7, 62]. Also in question are claims of system autonomy if construction of the models requires sources external to the system itself [17, 62].

3.4 Artificial Life

A wider purpose of artificial life is to understand and replicate the general properties of living systems [4]. The objective is to identify simple rules from which the complexity of life emerges. The concepts are tested by encapsulating the rules within software and executing the associated dynamical systems to see what transpires. Generally these systems have led to many levels of emergent organization that were not part of the original specification.

However, especially with regard to behavioral systems, exemplified by cellular automata, *Boids* [46], and Braitenberg's [8] *Vehicles*, the rules defined input-output transitions between states in the conventional manner, with little consideration for the unpredictable nature of the world.

3.5 Dynamical Systems

Overlapping with the bottom-up approaches of BBR and ALife has been the formal consideration of life and behavior as the interaction of two dynamical systems, of the agent and the environment. Inherent in this perspective are the concepts of situatedness and embodiment. The objective of dynamical systems theory is to explore the space of possible trajectories of the agent-environment system to gain insight into patterns of behavior. The coupling between an agent and its environment is considered as a symmetrical circular relationship where the environment acts on the agent and the agent acts on the environment.

Although this dynamical systems perspective is widely recognized as legitimate and significant, the field is, relatively speaking, in its infancy; it lacks the understanding of integration with high-level cognitive tasks and also lacks the theoretical concepts and formal tools necessary for understanding the dynamics of the interaction of the agents and environment [5].

4 Perception-Based Robotics

PCT represents a revolution in the way of thinking and understanding the nature of behavior and intelligent systems. Applying this perception-based perspective to robotics systems overcomes many of the limitations, complexities, and problems associated with the conventional approaches.

4.1 Modeling

Controlling input rather than output means that there is no role for predictive models within a perception-based system [7]. This significantly reduces the complexity involved in building robotics systems, as it removes the requirement to explicitly represent the physics and dynamics of the world.

The traditional AI approach to the problem of catching a baseball, for example, may involve something like the following serial processing stages: perceiving the baseball and the surrounding environment by extracting information from the visual scene to compile a symbolic representation of the contents; building a model of the environment and the relative positions of the baseball and the fielder within it; computing the trajectory of the baseball [3]; predicting the future position of the ball and the direction and speed at which the fielder would need to run to be in the same position; and executing the task of moving to the catch position, which would involve computing and generating the precise muscle tensions [24] required to move the body through the environment. An approach such as that taken with the Justin robot [3] clearly relies heavily on complex computation, of a type that could be described as a priori computation, which requires internal knowledge of properties of the environment and of the physics of the behavior of objects within the world. This is in contrast to simple computation within the brain, such as integration and differentiation (the equivalent of addition and subtraction), which results from excitatory and inhibitory activation of neurons.

PCT shows that there is a much simpler and more parsimonious solution to the baseball-catching problem, which is also compatible with the uncomplicated nature of neural computation. The solution is simply the control of the perceptual input variables—the vertical and horizontal retinal optical velocities of the ball with respect to a fixed background—and not the output variables [29, 31]. In basic terms this involves moving the body relative to the ball until it is in a position where the perceived velocities of the ball on the retina are zero. Contrary to conventional perspectives [20, 24], PCT indicates that prediction is not required for chasing and catching objects. Control-based research shows that fielders often run in the direction opposite to the one that would be indicated by a prediction strategy [29] and that people, and animals, are quite capable of chasing objects, such as helicopters [49] and Frisbees [48], that do not follow trajectories as potentially predictable as a baseball.

The PCT approach comprises no objective modeling, no mapping, no physics knowledge, no specific output, no prediction, no (a priori) computation. It is not necessary for the system to "know," independently of the system itself, what actions will result in the desired perception. The actions and *direction* of change are an inherent part of a pre-organized control system and are driven by the system error. The particular control systems, though, do not exist in isolation, but are part of a wider group that has formed through long-term development in association with the lower-level systems, such as the relevant muscles. These associated systems already encapsulate the control of perceptual goals required for movement and for running behavior.

4.2 Goals

There is a clear difference between a goal (a desire for something to happen) and a prediction (a future state derived from a model). A prediction of it being sunny tomorrow is a result of the execution of a deterministic model. *Wanting* it to be sunny tomorrow is merely an expression of a *desired* future state, without any specification of how that state could, or should, be achieved. Behaviorally, the speed of a car is related to the angle of the foot on the throttle pedal. The expression of a desired speed includes no specification of what that angle should be. Even the actual process of achieving that goal never requires an angle specification; the angle is simply varied until the goal speed is realized. In contrast, the determination of the speed with a predictive system would need to specify the foot angle, as well as many other parameters that constitute the model. If, however, the car were then to go up a hill, which would affect the speed, the predictive system would then need to model this disturbance in order to compute a new angle of the foot. A perceptual control system, on the other hand, in the face of such disturbances, continues the *same* process of varying the output until the perceptual goal is reached, without the need to model anything or make predictions.

PCT extends this concept, of specifying goals rather than predictions, to all levels of behavior thus removing the need for prediction or models. Any form of behavior, no matter how complex, can be expressed as a set of goals and subgoals. Following the PCT principle of the control of *input*, all of these systems specify *perceptions* that need to be satisfied. It is only at the interface with the environment that actions take place, and even there they are not specific, but dynamic according to needs. At no point are predictions or specifications of output required.

Action specification systems, which may be successful in simulated or controlled environments, will not work for robots in real, uncertain environments [43], where the actual conditions encountered will not match those from which the action was computed. For example, applying a specific force to open a door of known mass will fail to open a door of a higher, unknown mass. But a method of increasing the force until a door is *perceived* to be opening can, in principle, open any door, and without needing to know its mass.

Of course, the particular subsystems that are employed are an outcome of an error-reducing learning process, which determines which subgoals are suitable for achieving a higher goal, such as learning that it is the throttle that contributes to reaching a higher speed rather than the brake.

4.3 A Subversive Methodology

The concept of the control of perceptual input subverts the conventional view of behavior as the outcome of input-output transformations, whether from a *stimulus-response* model or a *cognitive* model [7]. The differences sometimes may seem subtle, but the implications are profound.

There are, for example, some superficial similarities between the functions within a control system loop and finite state machines. And there are some special cases, of binary perceptions and state changes, where the functions could be modeled by finite state machines. Generally, however, finite state machines do not capture either the continuous-in-parallel nature of the variables within feedback control systems or, most importantly, the fact that it is the inputs to the systems that are being controlled and maintained at desired values, counteracting unpredictable disturbances, rather than the system comprising a set of transitions from input to output states.

This unique way of conceptualizing living (and artificial) systems as being perception-based rather than behavior-based is crucial to both our understanding of living systems and our ability to build intelligent machines in a way that is realistic and meaningful. The resulting methodology presented here demonstrates that artificial systems can be constructed in a simpler manner than with the traditional approaches, without the need to define and embody complex models of the external world. Rather than just presenting an improved approach to the development of artificial systems, PCT suggests that the conventional control-of-output approach is misconceived, fundamentally flawed, and mired in unnecessary complexity.

5 Key Attributes and Benefits

PCT defines a methodology with the potential of being a unifying theory for behavioral sciences, replacing previous approaches. In advancing the development and construction of artificial intelligent systems, the theory results in significant, beneficial implications. In summary, the characteristics and advantages of the contribution are:

Perceptual control. It is the perceptual *input* that is controlled, not the output. Output is varied in order to minimize the difference between a desired and a current state of the perception.

Simplicity. The basic unit of perceptual control systems has a very simple operation, of negative feedback control.

Universality. The basic perceptual control process is common to all levels and types of behavior.

Scalability. The arrangement of basic control units into an interdependent hierarchy results in a highly scalable architecture, of simple units.

Purpose. The structure of the basic process of negative feedback bringing perceptual input into line with a desired state provides the system with innate, internal motivation, a necessary characteristic of purposeful systems.

Adaptivity. The structure of the basic perceptual control unit is inherently adaptive. A disturbance to the perceptual input results in error, which in turn results in output that acts upon the perception, automatically canceling out the effects of the disturbance. This functionality enables a system to control and maintain its goal despite wide-ranging disturbances in dynamic, chaotic, and unpredictable environments.

Self-organization. More general and sophisticated adaptation arises out of the structural reorganization of the hierarchy as a consequence of an unsupervised learning process driven by the reduction of overall error.

Dynamical systems. The tightly coupled interaction of a perceptual control system with its environment is a situated, embodied, and dynamical system. Significantly, though, PCT reveals that the relationship is actually asymmetrical, not symmetrical. That is, the effects of the environment on the agent are not equal to the effects of the agent on the environment. The latter involves an amplification factor (due to acquired, stored energy) by which the agent can make the output of the environment match the reference goal of the agent [42].

Autonomy. As the goals of the system and the means by which they can be achieved are themselves embodied within a perceptual control system, it can be said to be truly autonomous.

No prediction. As the intentions of a system are composed of (perceptual) goals rather than commands, it is not necessary for a system to define in advance what the output should be. Rather, output is generated in real time according to the actual circumstances.

No models. As output is not predicted, world models are not required. Therefore, associated problems of model acquisition and complexity are non-issues. The case for predictive world models is based upon the invalid premise that output is a function of input.

Subjectivity. The conventional approaches, based upon models of physical properties of the world, are limited to what can be objectively extracted from the world. Perceptions, on the other hand, denote far richer *subjective* perspectives that do not exist independently of the perceiving system—the consequence being that perception-based systems can behave with respect to a more extensive set of goals than a purely mechanistic approach would allow (though not necessarily in a positive way, when those goals conflict with another's subjective goals).

Complexity. The power of HPCT and the complexity of behavior derive not from complicated objective models of the physical world, but from the control of high-level, sophisticated subjective perceptions, along with the multitude of lower-level perceptions on which they depend.

Advanced psychology. These PCT principles of the control of multilevel, subjective perceptions indicate how to design a scalable architecture for constructing artificial systems that are psychologically advanced. Rather than just dealing with entities (e.g., navigation parameters) related to the objective world, a perception-based robot could also behave with respect to the control of perceptions of *relationships* (e.g., "near"), *sequences* (e.g., language), *programs* (e.g., recipes), and, in principle, high-level, subjective or abstract concepts such as *principles* (honesty, justice) and *system concepts* (religion, government).

Artificial life framework. An approach to building artificial intelligent systems based upon the principles of perceptual control is an ideal candidate for a framework for artificial life. It addresses some of the key ALife challenges [4], of defining a simple, common process and a scalable, dynamical hierarchy and of explaining how complex and intelligent behavior emerges from the simplicity.

PCT is a parallel, distributed network of simple agents (the perceptual control unit) that exhibits "global" behavior. However, as agents in the hierarchy are not only dependent upon other, lower-level agents but also prescribe lower-level agents, PCT is a multi level, bottom-up *and* top-down approach that leads to behavior of a much more sophisticated nature than is usually seen in ALife systems.

ALife research has led to some interesting insights into self-organization and into how simplicity gives rise to complex behavior; however, it is recognized that [9, 23] the current ALife methodologies are not sufficient for fully explaining life in terms of general behavior.

The process of perceptual control holds the promise of providing new techniques and opening up new lines of research to significantly advance the discipline by furnishing what Brooks describes as the missing "new stuff" [9] of life: an organizing principle of behavior that differentiates living from non living systems.

6 The System

The system used to implement the PCT model is a Lego NXT robotic kit configured as a small rover (see Figure 2) with two motorized wheels for propulsion. The two sensors used for the main experimental model are an ultrasonic sensor, which gives a single scalar value indicating proximity, and a light sensor, which gives a single scalar value indicating light level. For the high-level perception demonstration the light sensor is also used, as well as a sound sensor also giving a single scalar value.

The control systems themselves have been developed within a bespoke PCT framework of Java functions and libraries, which can be executed remotely over a Bluetooth connection or uploaded to the NXT running the LeJOS (www.lejos.org) operating system. Although slower, the experiments were carried out remotely, for debugging purposes, but gave a sufficient iteration rate of around 100 ms.

The controller for the main experimental robot is a four-level perceptual control system hierarchy. The detailed configuration and pseudocode for the system are outlined in the Appendix.

7 Behaviors and Perceptions

Viewing the footage [58] of the main system in action, the following observed behaviors may seem apparent:

- complete rotation of the unit searching for the highest light value,
- partial rotation to fixate on the light target,
- movement towards the target, adjusting the orientation, if away from the target,
- repetition of a sequence of (above three) behaviors,
- avoidance of obstacles,
- recovery from being stuck in dead-end situations.

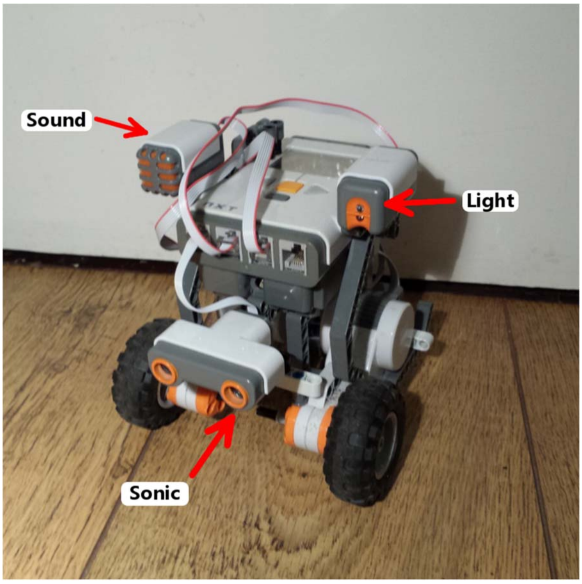

Figure 2. Lego NXT robotic rover. The ultrasonic sensor is at the front of the rover, and the light and sound sensors are on top. The dimensions of the unit are 14 × 18 × 20 cm. Processor specification: Atmel 32-bit ARM, 48 MHz, 256-KB Flash RAM, 64-KB RAM.

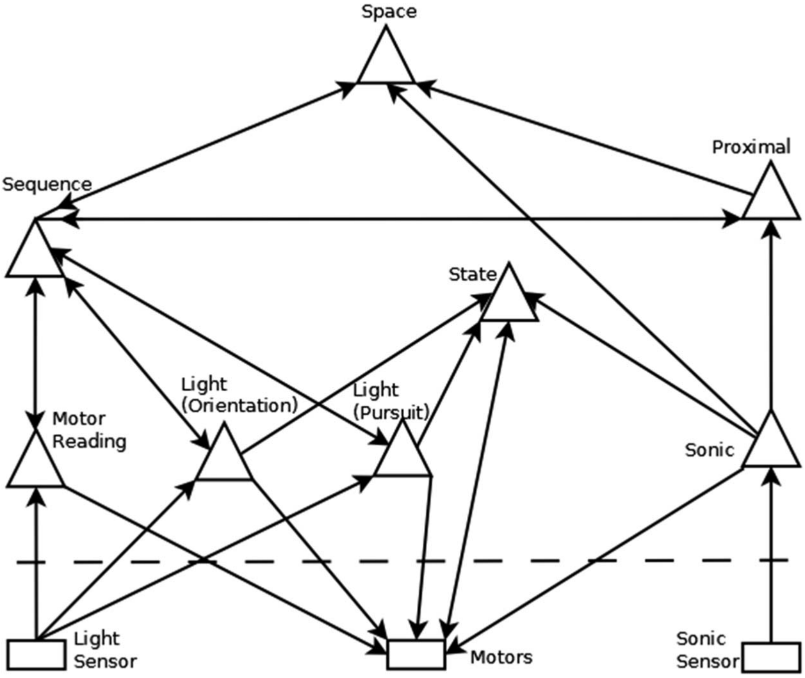

Figure 3. Main experimental model. A hierarchy of perceptual control units; for the sake of clarity some detail is omitted. The arrows indicate perceptions going up the hierarchy and references going down, depending upon the direction of the arrows. Note that some lines have double-headed arrows indicating both. Each triangle represents a basic perceptual control unit (see Figure 1), containing the perceptual, reference, comparator, and output functions.

However, none of these behaviors have been explicitly programmed into the system, and neither are there different solutions utilized for the different types of behavior, as per BBR. All the behaviors are the subjective observed outcomes (side effects) of a set of units sharing a common architecture, that of perceptual control systems.

The specific architecture of the implementation was selected through a process of manual organization and reorganization of the control systems, the connections between them, and the parameter settings, such as the gain, using the control of the goal variables as a measure of the validity of the configuration. With individual systems, settings were adjusted, ensuring that the goals were reached in a sensible time frame, and that they did not result in uncontrolled oscillations. It was found that, although there were limits, there was a wide range of values for settings that were acceptable for good control. In terms of the hierarchy it was generally judicious to work upwards from the lower levels, ensuring that those systems were suitably adjusted for successful and stable control before tackling the higher systems into which the lower systems would feed. In real systems the structure develops through a process of automatic reorganization.

The controlled perceptions from which the above behaviors emerge are as follows (described in more detail later; the words in brackets refer to the systems shown in Figure 3):

- control of perceived wheel movement, of both motors in opposite directions, by applying a speed to each wheel (Motor Reading),
- control of perceived light signal, with respect to the highest memorized value, by rotating the rover until the two correspond (Orientation),
- control of the transition of the perceived change in the signal, by modifying the relative weights applied to the two motors (Pursuit),

- control of the perception of a sequence of lower-level perceptions (previous three), by controlling each in turn (Sequence),
- control of the perceived ultrasonic signal, by relating the magnitude of the motors' speed to the error signal (Sonic),
- control of the sum of perceived changes of internal values, by reversing the motor weights (State).

Although there is a relation between the perceptions controlled and the observed behavior, it is not the behavior itself that is encoded within the implemented systems. Each of the systems merely controls a perception, without any concern for the consequences beyond its own boundary. In the first case, as far as the system itself is concerned, it just controls an input signal consisting of the amount of differential wheel movement, not of the rotation of the unit. The observed outcome is of a complete rotation of the unit, depending upon conditions. That the Rotation behavior is not encoded can be tested simply, by lifting the unit off the floor, which will result in the wheels turning until the wheel movement reference has been reached without the unit rotating. In terms of the control system, it has achieved its goal of moving the wheels the desired amount.

Each of these systems incorporates a common architecture of perceptual control systems; they differ only in the source of the perceptions that they control. Within a perceptual control system the emphasis is on the perceptions themselves; the behavioral output can be anything and is relevant only in that it enables the perception to be controlled.

8 The Model

The diagram in Figure 3 shows the control systems constituting the architecture of the main experimental system. Each triangle represents a basic perceptual control unit, containing the perceptual, reference, comparator, and output functions. Below the dashed line are the sensors and the actuators (motors). The motors are associated not only with outputs coming from the system, but also with inputs going into the system, by way of a tachometer sensor reading that is fed into control units to form perceptions.

Conventional approaches are concerned with reproducing and encoding *behaviors*, whereas here the concern is with the perceptions that are being controlled (from which observed behaviors emerge). So, the model systems and their names are related to, and describe, not behaviors, but the perceptions under control. This does require a different way of thinking about what behavioral systems are doing, especially as what are being encoded are internal processes and variables invisible to the observer. However, it does highlight that the conventional perspective of reproducing observed behavior has been neglecting essential properties.

A description of each of the control system units and how their interactions result in the overall observed behavior of the system, is as follows:

Space. The purpose of this highest-level system is to perceive being in open space, away from any objects. It does this by controlling a conditional relationship perception of not being close to an object, while the Sequence system is not active. If its goal is not to be perceived, then the lower-level Sequence system is initiated, which takes the rover into open space, thus satisfying the goal.

Proximal. This system also perceives the conditional relationship between the proximity of an object and the Sequence perception, but the reference specifies that the Sequence perception should *not* be active when close to an object. When it is perceived that the Sequence perception unit *is* active when the rover moves close to an object, the output changes, which, at a lower level, deactivates the Sequence system goal.

Sequence. The Sequence perception function takes as input the perceptual signals from the three lower-level control units. The resulting perception is a scalar sum of the three systems and represents the progression through the sequence. Each lower-level system only becomes active when the previous system in the sequence has achieved its goal. Figure 4 shows the control systems that constitute the sequence perception controller. The result of this control of sequence is that the robot is observed to first complete a full rotation, then turn back towards the light, and then move towards it.

Motor Reading. This unit controls the perception of the magnitude of the movement of the wheels, in opposite directions, perceived via the robot motors. Although it is not the rotation of the unit that is directly encoded in this system, the result, and the observed behavior, is that the rover performs a complete turn. It would have been possible to control horizontal angular degrees (with a different sensor) to produce a compete robot turn; however, the system that was chosen indicates that the perception under control is not directly related to the resulting observed behavior. While this is occurring, an independent node records the maximum light signal encountered, which is later used as a reference for other control units.

Light (Orientation). Using the memorized maximum light value as a reference, the rover turns until the current perceived light signal matches the reference. In other words, the rover turns to face the target.

Light (Pursuit). This controller is actually a two-level system. The top unit perceives and controls the perceived light value. The reference is the memorized maximum light value, and the control is achieved by setting positive speeds for both motors, moving the rover forward.

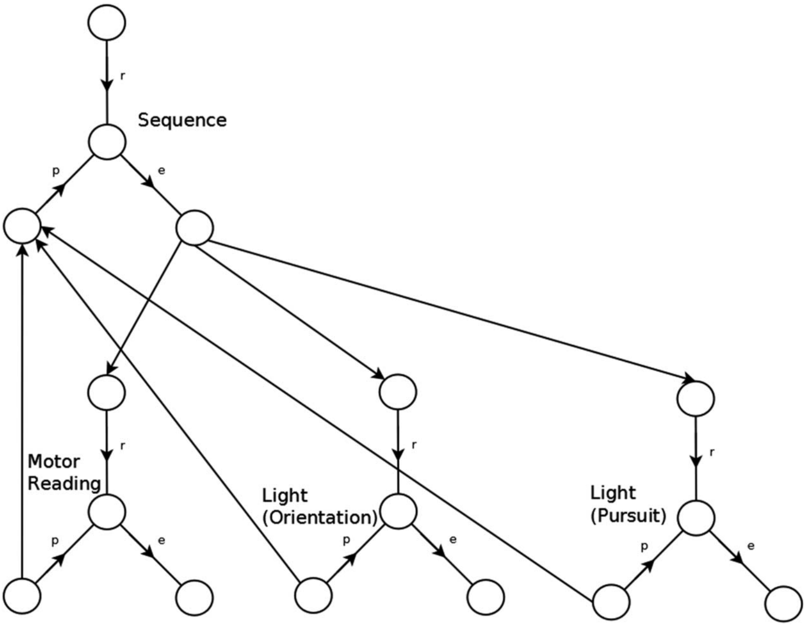

Figure 4. Sequence perception. The approximate architecture of a set of units able to control a perception of a sequence of events.

There are a number of disturbances that could result in losing the orientation, with respect to the target, such as differences in tires, the ground surface, or fluctuations in the light or movement of the target. However, the lower unit perceives and controls the *transition* of the *change* in the light signal as it passes from positive to negative, which occurs when the unit is moving away from the light source. With the goal of continually turning towards the light, an error signal only occurs when the change transitions negatively. The resulting output function reverses the weights applied to the motor outputs. The outcome is that the rover continually and dynamically turns back to the target as it moves forwards, despite disturbances and changes in the target light.

State. By a function that consists of a moving average of a sum of the *changes* within certain internal variables—namely, the light, sonic, and motor speed values—the rover is able to perceive when nothing is changing for an extended period of time. By changing the weights applied to the motors, which are output as a function of the error, this perceived situation can be controlled. Such control is useful when the rover gets into situations of local minima where the other control systems are not sufficient to perceive what is happening, or when they perceive erroneously.

Although this system has been constructed with deadlock recovery in mind, the goal of the system itself is to maintain continuous change within its incoming signals. It achieves this by changing its output, which in this case is connected to the motors. The *observed* behavior of the system is that it overcomes a deadlock situation.

Sonic. The reference for the sonic control unit is set at a fixed low value representing the closest it should get to an object. When the unit is applied in a comparator function with the current (perceived) ultrasonic reading, an error signal is generated, which the output function converts to a speed value, applied to the motors. So, as the rover gets closer to an object, it slows down until the reading and the reference are equal. If it overshoots, then the resulting speed is negative and the rover backs up.

The outcome, from the point of view of the observer, of the operation of these units is that the rover searches for a light target and moves towards it, avoiding obstacles. It is able to recover from deadlocks by monitoring, and controlling, its own internal state. As a whole system, it is also able to switch between high-level goals according to the circumstances, and can repeat sets of events in order to, in effect, reinitialize itself and its goals, enabling it to start afresh repeatedly and so continue indefinitely (battery permitting).

From this PCT architecture it can be seen that there is no central control, no knowledge representation, and no symbolic reasoning or processing. No one unit knows what any other is doing, and none has any need to know. They simply control their own perceptual inputs. However, from the connectedness and interaction of the units emerges a rich display of observed behavior, which belies the simplicity of the underlying principles.

In terms of HPCT, this particular instantiation of the architecture controls perceptions corresponding to about half the postulated categories of levels of the theory, at the lower end of the hierarchy, namely, intensities (light, sonic), transitions (light change), events (rotation), sequence, and programs (logical and conditional relationships).

It should be apparent that, in principle, such an architecture of perceptual controllers could account for much of the behavior observed in the animal kingdom. This is especially the case with sequence and program control, as behavior is often repetitive and conditional [25] on specific perceived events, and these two categories of perceptions controlled can result in an arbitrary combination of goals within a collection of perceptual control units.

It can also be seen that the variables controlled need not be scalar, but can be binary, and the whole system can flip from one state to another, giving the appearance of specific, distinct behavioral modes being generated, rather than the actuality of their being the by-product of bifurcations arising in the state of a dynamical system.

9 High-Level Perceptions

Perceptions exist only in the context of the architecture of a particular neural system and do not directly represent or correspond to objective properties or events in the external world. For example, when viewing a random dot stereogram (or a picture in an art gallery) it may be necessary to adjust one's perspective (focus or viewing position) in order to form a recognizable perception of what is being viewed. In this instance, the world elements themselves, on which the perception depends, do not change, but the internal variables, which combine to form the perception, do.

In behavior arising from perceptual control it is these internal constructions that are controlled, and not the external world; though there may also be manipulation of the physical world. This can be demonstrated by creating a simulated neural architecture that is able to perceive a *bowie* (a perceptual signal combining both sound and vision) and control the perception by changing the position of the rover in which the architecture is embodied.

The basic form of the perception is shown in Figure 5. By combining signals sourced from two different sensors, along with threshold functions, a perceptual signal arises only when sound and light are present at the same time and in the same direction, relative to the rover. In the environment, though, there is no such thing as a bowie *signal*, only the independent, primitive sources of photons and air vibrations. The bowie perception is entirely an internal construction of the neural architecture in which it arises.

With the same transition control system described earlier, though with the bowie as input rather than just light, the rover is able to locate (monitor and control) a bowie-related perception (see video [57]). This simple experiment, which uses basic threshold and AND functions within the context of a perceptual control system, demonstrates the conceptual point that perceptual input that has no direct correlate in the external world is controlled. In the real world, if the combination of events that make up the bowie corresponds to an energy source, then a bowie-perceiving creature would have, in a competitive environment, an evolutionary advantage over the others, and in a noncompetitive environment would be a more efficient energy acquirer, enabling it to spend more time on reproduction. The bowie-perceiver has no need to build a model of the world, but selectively attends only to those aspects of the environment that are relevant to its goal, and it acts to achieve that goal.

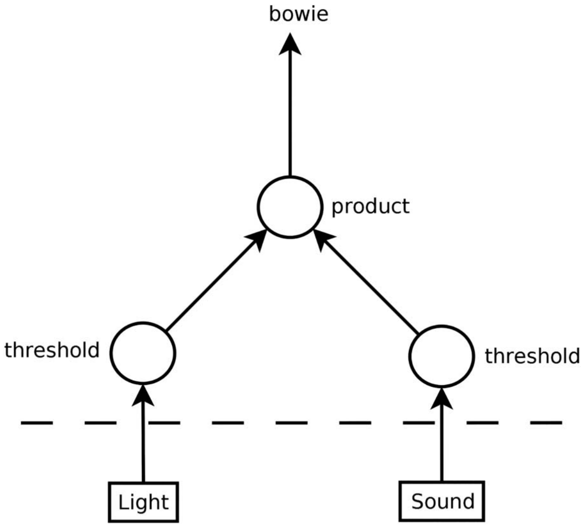

Figure 5. The bowie input function. The bowie is a thresholded combination of perceptions from two distinct sources, sound and vision.

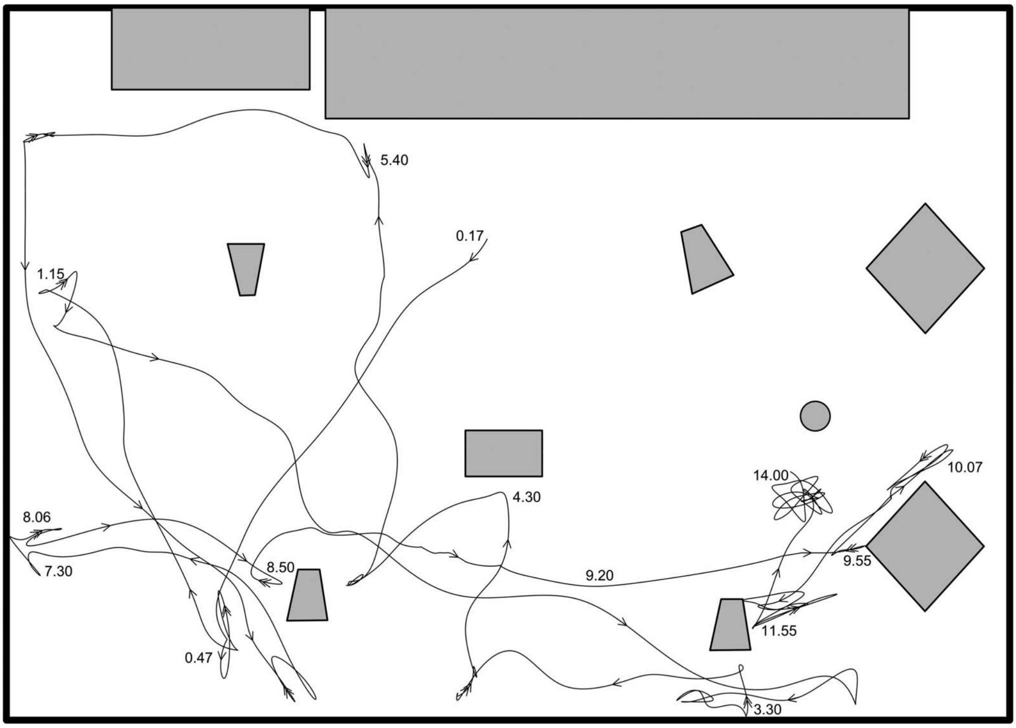

Figure 6. Rover route. The line shows the approximate route of the autonomous rover through the "laboratory" environment of the main experimental demonstration. The arrows show the direction of movement; the double arrows indicate reversal. Generally, changes of direction follow execution of the sequence perception system with the rover rotating to locate the light target. The numbers show the time along the route, corresponding to the video timings [58]. The shaded areas are the obstacles, and the circle is the light source switched on at time 8.50.

10 Results

The outcome of the implementation of a PCT architecture of control systems in a real-world scenario is demonstrated in the video (long version [58]) of the main experiment. Some particular points of interest, indicated by time, are listed below, followed by a fuller discussion of the key concepts demonstrated. The naive observer view is shown, along with a more detailed explanation of what is actually going on, with reference to the underlying model, as described in Section 8. Also, Figure 6 shows the trace of the path of the rover through the experimental environment.

[0.08] Rover rotates anticlockwise

Within the control of the sequence, the rover controls for the perception of the opposite wheel motion distance, resulting in the rover turning around. While this is happening, and at all other times, a *memory* node records the highest light value perceived from the light sensor; until a specific light source is introduced later, it is the value of the ambient light.

[0.17] heads off in direction of light target

As the orientation of the rover at the end of the rotation event happens to be in the direction of the light target, the goal of the second event in the sequence, turning to locate the target, is already realized, so the rover goes directly into the third event and moves forward towards the target.

[0.40] slows down as it comes close to a wall

The reference speed for the motors is a function of the proximity perception error.

Therefore, as the rover gets closer to an object, the error—the difference between the reference and perceived (current) proximity readings—reduces, as does the speed. The rover controls the perceived sonic signal.

[0.47] moves backward and forward briefly

As negative error (indicating closer than the reference) results in negative motor speed (indicating backwards), and a minimum nonzero speed is configured; the rover will move backward and forward around the reference proximity value. In parallel with this behavior, the Space control system perceives the to-and-fro behavior, by a moving average of the sonic error. This perception is controlled by triggering the sequence system into activity, the outcome of which is that the moving-average perception is brought to match a reference.

[0.50] rotates anticlockwise

Following from the previous point, the sequence perception is initiated, as is this event, the first in the sequence (see [0.08]).

[0.57] rotates clockwise

Once the full rotation event is complete, this second event in the sequence is initiated. The rover turns until the perceived light value matches the reference value, recorded during the previous event. The output of this system is simply the application of weights, of opposite signs, to each of the motors, with a constant speed value.

[1.00] heads off in direction of light target

Once the rover has aligned with the light target, bringing to a close the second event, the third is initiated. With the two-level Pursuit system, described earlier, controlling for the light value and the transition change, the rover continually corrects its orientation with respect to disturbances and a dynamically changing target.

[1.15–3.05] repeating pattern of perceptual control behavior

Repeated instances of control of sequence, proximity, internal state, and conditional perceptions.

[3.05] rover too close to wall and gets stuck

As the angle of approach to the wall is too acute, the rover ends up with the sonic sensor against the wall, so that the sonic echo is not received by the sensor and it returns the maximum value, indicating nothing proximal. The motors continue to turn, but the rover is blocked by the wall.

[3.30] rover reverses

With a perceptual function composed of the sum of the moving averages of the magnitudes of changes in the readings of the light and sonic sensors and of the motor speeds, the perceptual signal can indicate if nothing has changed for a while, generally indicating the the rover is stuck. Utilizing a reference (binary value) indicating the goal of perceiving change, an error signal is produced that reverses the motors, which has the effect of introducing changes in the inputs to this system's perceptual function and so achieving its goal (controlling its perception).

[3.40–8.06] repeating pattern of behavior

Repeated instances of control of sequence, proximity, internal state, and conditional perceptions.

[8.06] hits wall, but still reverses

As the angle of the wall in relation to the line of sight of the sonic sensor becomes greater than a right angle, the echo returned will be weaker, thus giving erroneous readings, in which case the proximity control may be loose, resulting in overrunning to the wall. However, in this case the reading is still valid enough to recover and reverse away from the wall.

[8.06–8.50] repeating pattern of behavior
Repeated instances of control of sequence, proximity, internal state, and conditional perceptions.

[8.50] new light source introduced
A new light source is introduced with the switching on of a spotlight. As the node that is recording the maximum light value is continually monitoring the light around it, it picks up the new light source, and the resulting value of the light becomes the new reference value for the location and pursuit events.

[9.20] moves towards light while correcting alignment
The new light source is now the target, and the rover moves across the room towards it, correcting its alignment to the light as it moves, as per the [1.00] description.

[9.43] snags on an object
The sonic signal bounces off the angular object at right angles to the faces of the object, and hence, due to the angle of approach, not back to the sensor. Consequently the rover runs straight into the obstacle.

[9.55] recovers from being stuck
As per [3.30], the rover perceives that its internal state is not changing and acts to change that perception.

[10.07–11.55] repeating pattern of behavior
Repeated instances of control of sequence, proximity, internal state, and conditional perceptions.

[11.55–14.00] remains close to target
As the rover is near the highest light value, it oscillates between the second and third events in the sequence, because each time it moves towards the light, the sensor moves out of the brightest area and the perceived value drops, so it rotates to reorient itself.

The implementation of the controller of this robotic unit as a perceptual control system highlights a number of key differences and advantages when compared to the existing approach of BBR [10].

- The process of perceptual control provides continuous assessment of the success of progression towards any goal. For example, in the experiment the Motor Reading system continually evaluates the degree of wheel movement, and the unit continues turning until the goal is reached—as is the case with the Light (Pursuit) system—which demonstrates counteraction of the effects of unknown disturbances inherent in the robot's real-world physical configuration as a consequence of the imperfect transmission system and uneven wheel balance on the trajectory of the rover.

  The BBR approach, in contrast, computes an output action to be performed from an input and, generally, executes the associated command to perform the action and waits until it is finished, without any evaluation of the consequences of the action, so it cannot compensate for any disturbances while the action is taking place. However, one module, Straighten [10], does come close to the perceptual control concept in that after each turning action it checks the orientation against a desired orientation and initiates another turning action if they do not match.

- All of the control units share the same, common process, of perceptual control, irrespective of what that perception might be seen to represent from an external observer's point of view. The significant benefit of this is that the central process can easily be replicated endlessly throughout any system, which enables the general architecture described here to be defined. The BBR modules, on the other hand, are semantically and functionally distinct, requiring bespoke implementation for each behavior.

- The actions generated by the rover control systems are functions of the goal evaluation and not computed as a function of the input, as is the case with the BBR system. The inputs to the systems are prone to undesirable influences, whether they be external disturbances or internal noise. So, it will be the case that the same output action should occur even in cases where the inputs are different. Therefore, an input-output methodology, as with BBR, cannot compute the correct outputs in anything but controlled conditions. The perceptual control approach overcomes this flaw by considering perceptual states as the goal. The result is that anything that unduly affects the goal can be counteracted by generating appropriate output on the fly.

- The perceptual inputs in the experimental hierarchy are more sophisticated at the higher levels, as they are derived from functions of sets of lower-level perceptions. The State system, for example, controls the perception of an internal state of dynamic change derived from a set of lower-level perceptions. This particular example is significant in that it does not represent a state of the external world, but an internal, subjective state. It actually highlights the fact that all perceptions are internal constructs (see Section 9) that do not exist independently of the system itself.

  Adding even more levels would result in the ability, in principle, to control ever more complex goals, due to what could be regarded as a richer representation of the relationship between the system and the world.

  The BBR approach, on the other hand, only considers low-order input variables that come directly from sensors. The absence of higher-order variables restricts the flexibility and scalability of the approach.

Table 1 summarizes some results from a set of subsequent runs [61] with the rover in a number of scenarios, starting at different distances from the light target, facing either towards or away from the target and either with a clear line of sight or with obstructions. In all cases the rover reaches the target, by various means. Generally, once the rover has acquired a clear line of sight to the target, it makes its way more or less directly to the target, compensating for the disturbances to its heading. If there is an obstruction, it initiates its sequence, which concludes with moving into open space. In only one case does the unit encounter a deadlock situation, and due to the control of the perception of internal state, it swiftly recovers. The rover may get caught in a local minimum near to an object where it takes a number of sequences to become clear (e.g., run 3). This is indicated by the correlation between the number of sequences and the time taken to reach the target. For example, in run 5, although the initial distance is the maximum, the target is acquired in one sequence and the unit quickly moves to the target. The tachometer value recorded is the sum of those of both motors, which may cancel each other out if the unit is rotating, as is the case with run 8, which has a high number of sequences, but low tachometer value.

The current system finds its way out of the local minimum due to the dynamics of the environment rather than by a configured strategy. The PCT methodology would solve this problem by adding another level, whereby it would be perceived that it is repeating control near to an object and could then take corrective action earlier than would otherwise be possible. This illustrates a general principle: that if a situation can be perceived, then it can, in principle, be controlled.

Although the routes taken may not be optimal, these experiments show that a system of simple perceptual controllers is reasonably robust in an environment of unpredictable effects on behavior, despite extremely simple sensors. To make the system more advanced, the solution may require

Table 1. Comparison of experimental runs.

Run	Time	Distance	Tacho	Towards	Direct	Sequences	Deadlocks
1	135	320	22651	Y	N	7	0
2	155	320	24037	Y	N	8	0
3	185	140	37198	Y	N	11	0
4	56	140	8105	Y	N	2	0
5	38	320	11954	Y	N	1	0
6	150	320	31636	N	N	7	0
7	75	140	9988	N	N	5	0
8	145	140	5986	N	N	9	0
9	106	300	26865	N	Y	4	0
10	44	260	8138	Y	Y	2	0
11	53	140	14968	N	Y	3	1
12	44	140	4487	N	Y	2	0
13	47	140	8803	Y	Y	2	0
14	107	140	14327	Y	Y	3	0

Notes. For each run, is given the time (seconds) to come within the reference distance of the target (≈30 cm), the initial distance to the target, the sum of the tachometer readings from both motors, the initial orientations facing towards or away from the target, whether the line of sight was clear or obstructed, the number of target search sequences initiated, and the number of times the rover encountered a deadlock situation.

more sophisticated sensors—not to build complex objective models, but to form higher-level perceptions that could be controlled.

11 Discussion and Conclusions

11.1 Behavior Is Control

The experimental models demonstrated here support the original hypothesis that behavior serves the control of perceptions, by showing how behavior can arise from the interactions of sets of simple perceptual control units with the real world, via sensors and actuators. The control systems do not generate specific output (or behavior), but *vary* the output in order to change the perceptions (input) to match what is desired (the goals). The continuous evaluation of perceptual states provides an ever-present measure of system goals, which must be perceptual in nature. The behavior exhibited is relatively complex given the crudeness of the sensors: the equivalent of two one-pixel sensors. With access to more sophisticated sensory systems there is greater potential for the complexity of the variants that could be perceived and thus controlled.

The incorporation of higher-level perceptions (combinations of lower-level perceptions) has also shown that behavior takes place with respect to the control of abstract perceptions, such as open space or Bowies. Extrapolating this perspective, systems could, in principle, control highly complex perceptions and abstract concepts [56, 63]. Those perceptual variables are at the top of, and depend upon, a pyramid of lower-level perceptions the foundations of which are solidly grounded in the environment in which the system operates.

Apart from memory being stored perceptions, the system as a whole and the individual control units contain no knowledge, or representation of symbols, but the model is able to function and to exhibit behavior in the real world, merely by the transfer and simple transformation of neural-like signals. Any representation perceived or described is strictly in the eye of the beholder, and not inherent within the modeled system.

11.2 Intelligence Is Control

More than just a different way of designing and building artificial systems, PCT raises issues fundamental to our understanding of how living systems function and to the nature and meaning of intelligence. The intimate coupling of perception and behavior as interdependent strands of continuous causal loops suggests that it is a mistake to consider them independently. Unfortunately, the early pioneers of AI did conclude that such a disembodied intelligence approach was valid [11], leading to research based upon computation involving knowledge representation and Search within unrealistic, virtual environments far from the complexities of the real world.

If the PCT view is valid, then intelligence is achieved precisely by not executing optimal algorithms within a realm of complete knowledge. What is actually clever about animals and humans is that they are able to behave (at all) in chaotic, unconstrained environments, bombarded by potential information and with very little knowledge indeed. They are able to do so by picking out (perceiving) only what is important to them, that which corresponds to their goals, and they are able to act in the environment to control those goals. In this context intelligence could be seen as the ability to control perceptions, and is relative to the complexity (and abstraction) of the perceptions that can be controlled. It transpires, then, that the philosophy, approach, and techniques of traditional AI are the antithesis of intelligence.

The progression to an ALife perspective did represent a move in the right direction, towards the goal of understanding intelligence, but whereas BBR is close to the *spirit* of biological systems [11], PCT is in the spirit *and* the *letter* of biological systems. PCT provides to those interested in behavior and cognition, intelligent or otherwise, a language (the main concept being controlled perceptions) by which purposive behavior and cognition can be understood and described, and an architecture with which purposeful behavior and cognition can be modeled and implemented.

11.3 A New Way of Thinking

At first, the development of perceptual control systems may not seem as intuitive and clear-cut as the approach based upon the apparent simplicity of an input-output perspective. However, research directions should be governed by the realism and validity of a theory and not by any (apparent) ease of implementation. PCT does require a different way of thinking about behavioral systems, that of goals as perceptions to be realized, from the perspective of the system itself, and not just what is observed from an external point of view. Once the conceptual shift to the system-centered perspective has been made, it should actually make more sense and be more intuitive to view purpose, and the subsequent behavior, from this frame of reference, as it would be more apparent how systems themselves accomplish those goals. As PCT research and development matures, design principles and techniques are likely to emerge enabling the systems to be designed and constructed more easily and more intuitively.

11.4 What Next?

The current project is just the first step in a line of research to apply the main principles of PCT within the robotics paradigm by defining a generic architecture and demonstrating successful application with the experimental model. In order to promote the application of PCT to ALife and robotics, there are many suitable themes for a future research roadmap, including:

Learning and memory. The current experimental model makes only basic use of learning and memory, in that it records the perception of the maximum light value and replays it as a reference signal for the goal of the target location and pursuit systems. Although simple, this constitutes the basic form of memory within PCT, in that reference signals are values of retrieved perceptions, and all behavior consists of the reproduction of past perceptions [40].

The control systems in the model were constructed by hand, but this is unlikely to remain practical as systems become more sophisticated. Learning will be an integral part of future versions of this robotic architecture, both for optimization within single systems and for the automatic development of the organizational control structure. Memory will also be vital in order for the systems to be able to dynamically acquire new goals with respect to novel environments and to be able to replay and control those goals in the face of those previously unseen domains.

The model presented here is comparatively simple, and the obstacles ahead on future avenues of research are not trivial. Probably the main challenge is to determine and understand the complex biological perceptual functions, honed by eons of evolution and tuned through neural plasticity, that produce the perceptions to be controlled. If PCT is a valid reflection of real systems, then this does, at least, provide evidence that the approach is, in principle, scalable to highly complex systems. Recent progress in deep learning neural networks [21] indicates potential techniques for application to the automatic development of complex perceptual functions. Both PCT and deep learning embody a hierarchical structure of basic, primitive units, which self-organizes according to an error cost function. The main difference is that, in the case of PCT, the reorganization is driven by *control* error [44, Chapter 8] rather than classification error.

Simulation. Although the ultimate target environment for the development of artificial systems is the real world, along with all its unpredictability, there will be significant benefits in implementing and testing the processes and systems of perceptual control within a simulated environment. A number of perceptual control systems have already been illustrated in simulation, including arm control, crowd behavior, multidimensional self-organizing systems, adaptive control, and balancing control [44, 60] and baseball catching, coordination control, hierarchical control, and the behavioral illusion [32]. Extending this methodology to a whole-world simulation would allow perceptual control robot configurations to be rapidly tested and be partially optimized before letting them loose in the real world.

Visual perception. One of the main areas of AI research has been in computer vision. Although much progress has been made, it still largely takes the form of extracting features to construct a symbolic representation of a scene—one stage of the questionable perceive-plan-execute model. PCT may not reduce the complexities of the visual perception endeavor, but the context of PCT, concerned as it is only with information necessary to achieve specific goals, should indicate areas of research that are relevant and exclude those that are not.

Interactive robotics. The crowd behavior [44, Chapter 9] and collective control [34] work previously carried out in the context of PCT provide a clear and straightforward rationale for designing robots that interact with each other and with humans. The perspective provided by PCT of the perceptual goals controlled by the systems themselves provides a more coherent approach than attempting to infer rules of behavior as seen from the point of view of the external observer [26, 53]

Ethology. As higher perceptual control systems are dependent upon lower systems, it is inevitable that a bottom-up approach will be more conducive to progress than trying to implement high-level cognition without the underlying structure. Studies from the field of animal behavior, then, represent a potentially rich seam of existing knowledge that may be easily amenable to being reevaluated and expressed in terms of perceptual control. Investigating what animals perceive and how those perceptions are controlled may indicate a route to building up a hierarchy of control, which could be implemented with the current proposed architecture, from the lowest levels to higher levels, based upon simple to more complex organisms.

The *Boids* [46] simulation is a good example of a conceptualization of animal behavior that could easily be reconceived (and simplified) in terms of perceptual control. The three “behavioral rules” of *collision avoidance*, *velocity matching*, and *flock centering* would all be realized simply by the control of the perception of proximity to flockmates (by varying direction and velocity), as demonstrated by the similar PCT-based crowd simulation [44, Chapter 9].

11.5 Missing Stuff

PCT illustrates a novel, biologically plausible perspective on how living systems operate, and suggests a methodology and architecture for unifying the behavioral sciences [33, 13] and for the construction of artificial systems. The work described in this article has provided some validation of the theory and that methodology by its application to robotics and has presented an architecture that can be generalized to all artificial intelligent systems. As the architecture comprises simple units based upon the control of input rather than complex modeling for the control of output, there is significant potential for simplifying the construction, and improving the efficiency, of robotic systems.

The biological, evolutionary path represents the only known example that has produced intelligent beings. It is highly questionable whether other approaches are likely to be fruitful, given the context in which that intelligence has emerged. For a few decades it has been recognized that an approach to developing artificial intelligent systems will require situatedness, embodiment, and dynamism. What has not been recognized, until now, is that it will also require *perceptual control*.

References

1. Anderson, M. L. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, *149*(1), 91–130.
2. Aulinas, J., Petillot, Y., Salvi, J., & Lladó, X. (2008). The SLAM problem: A survey. In T. Alsinet, J. Puyol-Gruart, & C. Torras (Eds.), *Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence* (pp. 363–371). Amsterdam: IOS Press. URL http://dl.acm.org/citation.cfm?id=1566899.1566949.
3. Bäuml, B., Birbach, O., Wimböck, T., Frese, U., Dietrich, A., & Hirzinger, G. (2011). Catching flying balls with a mobile humanoid: System overview and design considerations. In *2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (pp. 513–520). New York: IEEE.
4. Bedau, M. A. (2003). Artificial life: Organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*, *7*(11), 505–512.
5. Beer, R. D. (2014). Dynamical systems and embedded cognition. In K. Frankish & W. Ramsey (Eds.), *The Cambridge handbook of artificial intelligence* (pp. 128–149). Cambridge, UK: Cambridge University Press.

6. Bench-Capon, T., & Dunne, P. E. (2007). Argumentation in artificial intelligence. *Artificial Intelligence*, *171*(10–15), 619–641.

7. Bourbon, W. T., & Powers, W. T. (1999). Models and their worlds. *International Journal of Human-Computer Studies*, *50*(6), 445–461.

8. Braitenberg, V. (1986). *Vehicles: Experiments in synthetic psychology*. Cambridge, MA: MIT Press.

9. Brooks, R. (2001). The relationship between matter and life. *Nature*, *409*(6818), 409–411. URL http://dx.doi.org/10.1038/35053196.

10. Brooks, R. A. (1985). A robust layered control system for a mobile robot (Tech. rep.). Cambridge, MA: Massachusetts Institute of Technology. URL http://portal.acm.org/citation.cfm?id=888753.

11. Brooks, R. A. (1991). Intelligence without reason. In J. Myopoulos & R. Reiter (Eds.), *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)* (pp. 569–595). San Mateo, CA: Morgan Kaufmann. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.2973.

12. Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139–159. URL http://citeseer.ist.psu.edu/brooks91intelligence.html.

13. Carey, T. A., Mansell, W., & Tai, S. J. (2014). A biopsychosocial model based on negative feedback and control. *Frontiers in Human Neuroscience*, *8*(94). URL http://www.frontiersin.org/human_neuroscience/10.3389/fnhum.2014.00094/abstract.

14. Chaumette, F., & Hutchinson, S. (2006). Visual servo control, part I: Basic approaches. *IEEE Robotics and Automation Magazine*, *13*(4), 82–90.

15. Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: Part I. *IEEE Robotics Automation Magazine*, *13*(2), 99–110.

16. Dyhr, J. P., Morgansen, K. A., Daniel, T. L., & Cowan, N. J. (2013). Flexible strategies for flight control: An active role for the abdomen. *The Journal of Experimental Biology*, *216*(9), 1523–1536. URL http://jeb.biologists.org/content/216/9/1523.abstract.

17. Gomes, L. (2014). *Hidden obstacles for Google's self-driving car*. http://www.technologyreview.com/news/530276/hidden-obstacles-for-googles-self-driving-cars/. (Accessed: January 2016).

18. Guizzo, E. (2011). *How Google's self-driving car works*. http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works. (Accessed: January 2016).

19. Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, *42*(1–3), 335–346. URL http://arxiv.org/html/cs.AI/9906002.

20. Hawkins, J., & Blakeslee, S. (2004). *On Intelligence*. New York: Times Books.

21. Hinton, G. E. (2009). Deep belief networks. *Scholarpedia*, *4*(5), 5947. revision 91189.

22. Jekely, G., Colombelli, J., Hausen, H., Guy, K., Stelzer, E., Nedelec, F., & Arendt, D. (2008). Mechanism of phototaxis in marine zooplankton. *Nature*, *456*(7220), 395–399. URL http://dx.doi.org/10.1038/nature07590.

23. Johnston, J. (2010). *The allure of machinic life*. Cambridge, MA: MIT Press.

24. Jordan, M. I., & Wolpert, D. M. (1999). Computational motor control. In M. Gazzaniga (Ed.), *The cognitive neurosciences* (pp. 601–620). Cambridge, MA: MIT Press.

25. Keijzer, F. (2012). The sphex story: How the cognitive sciences kept repeating an old and questionable anecdote. Available at https://www.academia.edu/4034267/The_Sphex_story_How_the_cognitive_sciences_kept_repeating_an_old_and_questionable_anecdote. (Accessed: January 2016.)

26. Koene, A., Rank, M., Roberts, R., Endo, S., Cooke, J., Mademtzi, M., Miall, C., & Wing, A. M. (2014). Coglaboration: Towards fluent human-robot object handover interactions. In M. Mistry, A. Leonardis, M. Witkowski, & C. Melhuish (Eds.), *Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1–3, 2014. Proceedings*, *vol. 8717* (p. 265). Berlin, Heidelberg: Springer.

27. Levitin, D. J. (2002). Experimental design in psychological research. In D. J. Levitin (Ed.), *Foundations of cognitive psychology* (pp. 115–130). Cambridge, MA: MIT Press.

28. Luo, L., Li, P., & Wang, H. (2012). Vehicle adaptive cruise control design with optimal switching between throttle and brake. *Journal of Control Theory and Applications*, *10*(4), 426–434. URL http://dx.doi.org/10.1007/s11768-012-0319-0.

29. Marken, R. S. (2001). Controlled variables: Psychology as the center fielder views it. *The American Journal of Psychology*, *114*(2), 259–281.

30. Marken, R. S. (2002). Looking at behavior through control theory glasses. *Review of General Psychology*, *6*(3), 260–270.

31. Marken, R. S. (2005). Optical trajectories and the informational basis of fly ball catching. *Journal of Experimental Psychology. Human Perception and Performance*, *31*(3), 630–634.

32. Marken, R. S. (2015). *Demos*. Available at http://mindreadings.com/demos.htm. (Accessed: January 2016).

33. Marken, R. S., & Mansell, W. (2013). Perceptual control as a unifying concept in psychology. *Review of General Psychology*, *17*(2), 190–195.

34. McClelland, K. (2004). The collective control of perceptions: Constructing order from conflict. *International Journal of Human-Computer Studies*, *60*(1), 65–99.

35. Minsky, M. (1963). Steps toward artificial intelligence. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought* (pp. 406–450). New York: McGraw-Hill.

36. Nasisi, O., & Carelli, R. (2003). Adaptive servo visual robot control. *Robotics and Autonomous Systems*, *43*(1), 51–78.

37. Nielsen, M. A. (2015). Neural networks and deep learning. Available at http://neuralnetworksanddeeplearning.com/. (Accessed: January 2016.)

38. Nieuwenhuis, S., & de Kleijn, R. (2013). The impact of alertness on cognitive control. *Journal of Experimental Psychology: Human Perception and Performance*, *39*(6), 1797–1801.

39. Pellis, S. M., & Bell, H. C. (2011). Closing the circle between perceptions and behavior: A cybernetic view of behavior and its consequences for studying motivation and development. *Developmental Cognitive Neuroscience*, *1*(4), 404–413.

40. Powers, W. T. (1973). *Behavior: The control of perception*. Hawthorne, NY: Aldine DeGruyter.

41. Powers, W. T. (1978). Quantitative analysis of purposive systems: Some spadework at the foundations of scientific psychology. *Psychological Review*, *85*(5), 417–435.

42. Powers, W. T. (1989). The asymmetry of control. In W. T. Powers (Ed.), *Living control systems* (pp. 251–252). Gravel Switch, KY: The Control Systems Group.

43. Powers, W. T. (1998). *Making sense of behavior: The meaning of control.* New Canaan, CT: Benchmark Publications.

44. Powers, W. T. (2008). *Living control systems III: The fact of control.* Bloomfield, NJ: Benchmark Publications.

45. Prinz, J. (2006). Beyond appearances: The content of sensation and perception. In T. S. Gendler & J. Hawthorne (Eds.), *Perceptual experience* (pp. 434–461). Oxford, UK: Oxford University Press.

46. Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Computer Graphics*, *21*(4), 25–34. URL http://dx.doi.org/10.1145/37402.37406

47. Roberts, L. G. (1963). *Machine perception of three-dimensional solids*. Ph.D. thesis, MIT, Cambridge, MA.

48. Shaffer, D., Krauchunas, S., Eddy, M., & McBeath, M. (2004). How dogs navigate to catch Frisbees. *Psychological Science*, *15*(7), 437–441.

49. Shaffer, D. M., Marken, R. S., Dolgov, I., & Maynor, A. B. (2013). Chasin' choppers: Using unpredictable trajectories to test theories of object interception. *Attention, Perception, & Psychophysics*, *75*(7), 1496–1506.

50. Stephens, D. W., Brown, J. S., & Ydenberg, R. C. (2008). *Foraging: Behavior and ecology*. Chicago: University of Chicago Press.

51. Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(2), 257–285.

52. Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460. URL http://dx.doi.org/10.2307/2251299.

53. Vijayakumar, S. (2014). Robots that learn: The future of man or the 'man of the future'? Available at http://www.perceptualrobots.com/2014/09/18/taros-2014-iet-public-lecture. (Accessed: January 2016).

54. Yin, H. H. (2014). How basal ganglia outputs generate behavior. *Advances in Neuroscience*, *2014*. URL http://dx.doi.org/10.1155/2014/768313.

55. Young, D. (1989). *Nerve cells and animal behaviour.* Cambridge, UK: Cambridge University Press.

56. Young, R. (2000). *Visual control in natural and artificial systems.* Ph.D. thesis, Centre For Vision, Sound and Signal Processing, University of Surrey, Guildford, UK.

57. Young, R. (2014). Bowie control. Available at http://youtu.be/TzxkDNCDVZ0. (Accessed: January 2016.)

58. Young, R. (2014). HPCT autonomous robot. Available at http://youtu.be/xtYu53dKz2Q. (Accessed: January 2016.)

59. Young, R. (2015). HPCT autonomous robot (short version). Available at http://youtu.be/JM9tSYpeLZA. (Accessed: January 2016.)

60. Young, R. (2016). Crowd behavior. Available at https://youtu.be/SR_hZw2_5YI. (Accessed: February 2016).

61. Young, R. (2016). HPCT experiment runs. Available at https://youtu.be/OYfqRq8KnPM. (Accessed: February 2016.)

62. Young, R. (in press). Robotics in the real world. In W. Mansell & A. McElhone (Eds.), *Living control systems IV.* Benchpress Publications.

63. Young, R., & Illingworth, J. (1997). A fixation and viewpoint measure for object-based gaze control. In A. F. Clark (Ed.), *British Machine Vision Conference* (pp. 570–579).

Appendix

A.1 Experiment Configuration

The details of the configuration of the main experiment are shown in Table 2. The control system comprises a set of perceptual controllers at a number of different levels. It is not necessary to implement the system in its entirety in order to observe behavioral functionality. Each controller, and its dependent lower levels, can be implemented and tested in isolation to demonstrate what it is doing. Each controller consists of at least one of input, reference, error, and output functions. Each of these may have a set of associated transfer functions.

In the table are shown, for each function:

- The name used to indicate connections between functions.
- The type, specifying a software function to execute. The pseudocode details are in Appendix A.4.
- A description of the function.
- A list of parameters, delimited by ';'. Each parameter is given its name and value, delimited by ':'.
- A list of links representing connections from other functions, which constitute the inputs to the function.

All the data transferred between functions are scalar signals, not messages containing symbolic information. Each function takes one or more inputs and produces a single output. The software framework takes the configuration and executes each function continually in a loop. For the main experiment the iteration rate was approximately 100 ms.

N.B.: In the specification the names of the controllers Motor Reading, Light (Orientation), and Light (Pursuit) are called, respectively, RotationControl, LocationRotationSensorControl, and PursuitControl.

A.2 Processing Order

Although it may be possible to execute the system in a parallel or multithreaded environment, the current framework executes each function serially, in the order outlined here.

A.2.1 Functions

For each controller the main functions were processed in the order of reference, input, error, and output. The associated transfer functions are processed in order—before the main function for reference and input, and after for output. So, for example, for the LocationRotationSensorControl controller the order of processing of the functions is LocationRotationReferenceWeighted, LocationRotationReferenceGreaterThan, LocationRotationSensorControlReference, LocationRotationSensorControlInput, LocationRotationSensorControlError, LocationRotationSensorControlOutput, PartialSequenceControl Active, PartialSequenceControlActiveOpposite, LocationRotationWeightBSign.

A.2.2 Controllers

The controllers for this experiment were processed in the following order: Env, LightSensor, UltrasonicSensor, Max, MotorInput, NegativeTransitionControl, PursuitControl, SequenceControl, SpaceControl, RotationControl, LocationRotationSensorControl, SonicControl, SensorSumChangeControl, MotorOutput, SequenceSwitch.

A.3 Design Considerations

Listed here are some brief comments about the principal points to consider when designing any perceptual control system, with the first, the controlled variable, being key.

The controlled variable. The perceptual, or input, signal, which is affected by the output of the system. It could be sourced directly from a sensor or could be a higher-level variable derived from other perceptions, such as a derivative or a weighted sum. The selection of the variable to be controlled is crucial, as it is fundamental to what the system actually does, but may not be obvious from the observed behavior.

The goal. The reference signal defining the desired value of the controlled variable. This signal may itself be dynamic as set by the varying output of a higher system.

The output. Any output of a system is relevant only if it, in some way, affects the controlled variable. That may be directly, through the environment, or indirectly, by affecting the reference of a lower system.

The hierarchy. Inherent in the perceptual control methodology is the notion that any behavior can be decomposed into a set of goals, each of which represents a perceptual control system. The design of the hierarchy should reflect this idea and identify the subgoals that contribute to the overall purpose of the system.

In testing the hierarchy it is generally judicious to work upwards from the lower levels, ensuring that those systems are suitably adjusted for successful control before tackling the higher systems into which the lower systems feed.

A.4 Function Definitions

The function types from Table 2 are defined below. Unless specified otherwise, y is the current input to the function, and x is the new output and the output value from the previous iteration.

MedianFilter Returns the median value of a set of values extending into the past; the set length is defined by the size parameter.

NXTUltrasonicSensor The raw value of the ultrasonic sensor signal.

Smooth An exponential smoothing of a signal.
$x = y_t \cdot (1 - s) + y_{(t-1)} \cdot s$, where s is a smoothing weight between 0 and 1, and t denotes the current iteration.

LightSensor The raw value of the light sensor signal.

Change The change in signal value between iterations.

DelayPeriod The signal value delayed by n iterations; default of n is 1.

DigitalLimit Returns one of two configured values, if the input is above or below a threshold,

$$\text{if}(y > t) x = u \;\; \text{else} \;\; x = l$$

where t is the threshold, u is the upper value, and l is the lower value.

MotorRead The current motor speed.

MotorWrite Sets the desired speed of the motor.

WeightedSum The weighted sum of a set of inputs.

WeightedMultiply The weighted product of a set of inputs.

Subtract The difference between two inputs, $x = a - b$.

PassThrough No transformation of signal.

Flip Flips a signal between one of two values when the input is positive:

$$\text{if}(y > 0)\{\text{if}(x == a) x = b \;\; \text{else} \;\; x = a\}$$

where a and b are the two configured values.

TimeIntegral The integral of a signal as a function of time:

$$x = x + y \cdot dt,$$

where dt is the iteration time in milliseconds.

LimitBanded Returns one of three configured values, depending on whether the input is above or below a banded range:

$$\text{if}(y > t + r)\{x = u\} \;\; \text{else} \;\; \text{if}(y < t - r)\{x = l\} \;\; \text{else} \;\; \{x = b\}$$

where r is the range, u is the upper value, l is the lower value, and b is the band value.

ZeroFilter If the input is zero, the output is set to a configured value; otherwise, the output equals the input:

$$\text{if}(y == 0) x = v \;\; \text{else} \;\; x = y$$

where v is the configured value.

ChangeSign Changes the sign of a signal.

GreaterThan Returns one of two configured values if one input is greater than another input:

$$\text{if}(a > b)x = u \;\text{ else }\; x = l$$

where a and b are the input values, u is the upper value, and l is the lower value.

Constant A constant value.

Scaling A scaled value of the input with a configured offset,

$$x = sgn \cdot o + y \cdot s$$

where sgn is the sign of the input, o the offset, and s the scaling factor.

Habituate A signal that habituates at a specified rate:

$$\text{if}(y == 0 \;\text{ and }\; x == 0)x = 0 \;\text{ else }\; \{\text{if}(x == 0)x = y \;\text{ else }\; x = x - r\}$$

where r is the rate.

Maximum The maximum value of a signal, which attenuates if a rate is set:

$$\text{if}(y > x)x = y \;\text{ else }\; \text{if}(r \;\, != null)x = x \cdot r$$

where r is the rate.

A.5 Pseudocode

If the systems for the main experiment were described in a symbolic, natural-language manner, the behavior could be captured in the following pseudocode. It is important to note, however, that the procedure "commands" are not specifying actions to be performed, but are specifying what is to be *perceived.*

```
procedure: main
  if perceiving close to an object and not perceiving sequence
    perceive sequence
  else
    deactivate sequence

procedure: sequence
  perceive rotation
  perceive locate
  perceive pursuit

procedure: rotation
  while tachometer distance perceived $< 1000$
    engage motors with opposite weights

procedure: locate
  while light perceived $<$ max
```

```
    engage **motors** with opposite weights

procedure: **pursuit**
  **while** light perceived $<$ max
    engage **motors** with biased weights
    **if** zero-crossing light change perceived
      flip weights

procedure: **sonic**
  **while** proximity perceived $\neq 20$
    $speed = \mathrm{sgn}(e)(x + 3 \cdot \mathrm{abs}(e))$

procedure: **motors**
  **while** speed perceived $\neq$ reference speed
    $speed = speed + f(e)$
```

Table 2. Control system specification.

Level		0		
Controller		UltrasonicSensor		
		InputFunctions		
Input	Name	USensorMedian	Type:	MedianFilter
	Description	Median value of sonic sensor		
	Parameters	Size:3		
	Links	USensorRead		
Transfer	Name	USensorRead	Type:	NXTUltrasonicSensor
	Description	Value of sonic sensor		
	Parameters	SensorPort:S1		
Controller		LightSensor		
		InputFunctions		
Input	Name	SensorSmooth	Type:	Smooth
	Description	Smoothed value of light sensor		
	Parameters	Smoothness:0.1		
	Links	LightSensorRead		
Transfer	Name	LightSensorRead	Type:	LightSensor
	Description	Value of light sensor		
	Parameters	SensorPort:S2;Floodlight:false		
Transfer	Name	SensorChange	Type:	Change
	Description	The change in light sensor value		
	Links	LightSensorRead		
Transfer	Name	SensorChangeDelay	Type:	DelayPeriod
	Description	Delayed change of sensor		
	Links	SensorChange		

Table 2. (*continued*)

Transfer	Name	FilteredChangeDelayed	Type:	DigitalLimit
	Description	A binary value indicating positive or negative change		
	Parameters	Threshold:–0.01;Upper:0;Lower:1		
	Links	SensorChangeDelay		
Transfer	Name	FilteredChange	Type:	DigitalLimit
	Description	A binary value indicating positive or negative change		
	Parameters	Threshold:–0.01;Upper:0;Lower:1		
	Links	SensorChange		
Controller		MotorInput		
		InputFunctions		
Input	Name	MotorBDirection	Type:	DigitalLimit
	Description	A value, 1 or –1, indicating motor direction		
	Parameters	Threshold:0;Upper:1;Lower:–1		
	Links	MotorBInput		
Transfer	Name	MotorBInput	Type:	MotorRead
	Description	Current motor speed		
	Parameters	MotorIndex:B		
		OutputFunctions		
Output	Name	MotorCInput	Type:	MotorRead
	Description	Current motor speed		
	Parameters	MotorIndex:C		
Transfer	Name	MotorCDirection	Type:	DigitalLimit
	Description	A value, 1 or –1, indicating motor direction		
	Parameters	Threshold:0;Upper:1;Lower:–1		
	Links	MotorCInput		

Table 2. (*continued*)

Controller		MotorOutput		
		InputFunctions		
Input	Name	MotorBOutput	Type:	MotorWrite
	Description	Setting value of motor speed		
	Parameters	MotorIndex:B		
	Links	SpeedReferenceB		
Transfer	Name	PursuitSpeedWeightB	Type:	WeightedSum
	Description	Speed weight of motor B for pursuit mode		
	Parameters	Weights:1,1		
	Links	PursuitDirectionWeight,ReverseWeight		
Transfer	Name	SpeedReferenceB	Type:	WeightedMultiply
	Description	Reference value for motor B		
	Parameters	Weights:1,1,1,1;Root:false		
	Links	SpeedReference,FullRotationWeightsBOutput		
		LocationRotationWeightsBOutput,Pursuit SpeedWeightB		
		OutputFunctions		
Output	Name	PursuitSpeedWeightC	Type:	Subtract
	Description	Speed weight of motor C for pursuit mode		
	Links	ReverseWeight,PursuitDirectionWeight		
Transfer	Name	SpeedReferenceC	Type:	WeightedMultiply
	Description	Reference value for motor C		
	Parameters	Weights:1,1,1,1;Root:false		
	Links	SpeedReference,FullRotationWeightsCOutput		
		LocationRotationWeightsCOutput,Pursuit SpeedWeightC		

Table 2. (*continued*)

Transfer	Name	MotorCOutput	Type:	MotorWrite
	Description	Setting value of motor speed		
	Parameters	MotorIndex:C		
	Links	SpeedReferenceC		
Level		1		
Controller		NegativeTransitionControl		
		InputFunctions		
Input	Name	NegativeTransitionControlInput	Type:	DigitalLimit
	Description	Generates a pulse if the change turns negative		
	Parameters	Threshold:0.5;Upper:0;Lower:1		
	Links	TransitionToNegativeChange		
Transfer	Name	TransitionToNegativeChange	Type:	Subtract
	Description	If the change has gone from +ve to –ve then this will output 1.		
	Links	FilteredChange,FilteredChangeDelayed		
		ReferenceFunctions		
Reference	Name	NegativeTransitionControlReference	Type:	PassThrough
	Description	A value of 1 indicates that the pursuit mode is active		
	Links	PursuitControlOutput		
		ErrorFunctions		
Error	Name	NegativeTransitionControlError	Type:	Subtract
	Description	Error of sensor		
	Parameters	Sign:1		
		OutputFunctions		
Output	Name	NegativeTransitionControlOutput	Type:	PassThrough
	Description	Output of transition		
	Links	NegativeTransitionControlError		

Table 2. (*continued*)

Transfer	Name	PursuitDirectionFlip	Type:	Flip
	Description			
	Parameters	ValueA:–0.2;ValueB:0.2		
	Links	NegativeTransitionControlOutput		
Transfer	Name	PursuitDirectionWeight	Type:	WeightedMultiply
	Description	Non-zero if pursuit control is active and sensors change is high		
	Parameters	Weights:1,1,1;Root:false		
	Links	PursuitDirectionFlip,PursuitControlOutput		
		SensorSumChangeRotationWeight		
Transfer	Name	ReverseWeight	Type:	PassThrough
	Description	Pass through		
	Links	SensorSumChangeReverseWeight		
Level		2		
Controller		RotationControl		
		InputFunctions		
Input	Name	RotationControlInput	Type:	TimeIntegral
	Description	A measure of how much the unit has rotated		
	Parameters	Tolerance:0.01		
	Links	MotorInput,RotationControlReference		
Transfer	Name	MotorRotationValue	Type:	WeightedSum
	Description	Rotation value, 0 if rotating		
	Parameters	Weights:1,1		
	Links	MotorBDirection,MotorCDirection		
Transfer	Name	MotorRotationValueWeight	Type:	LimitBanded
	Description	Binary rotation indication		
	Parameters	Threshold:0;BandWidth:0.5;BandValue:1;Upper:0;Lower:0		
	Links	MotorRotationValue		

Table 2. (*continued*)

Transfer	Name	MotorInput	Type:	WeightedMultiply
	Description	Speed if the unit is rotating and rotation control is active		
	Parameters	Weights:1,1,1;Root:false		
	Links	MotorBInput,RotationControlOutput MotorRotationValueWeight		
		ReferenceFunctions		
Reference	Name	RotationControlReference	Type:	PassThrough
	Description	The reference value for unit rotation		
	Links	FullRotationReferenceWeighted		
		ErrorFunctions		
Error	Name	RotationControlError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:1		
		OutputFunctions		
Output	Name	RotationControlOutput	Type:	DigitalLimit
	Description	Indication of whether rotating		
	Parameters	Threshold:0;Upper:1;Lower:0		
Transfer	Name	FullRotationWeightsBOutput	Type:	ZeroFilter
	Description	Ensures that the output weight for motor B is non-zero		
	Parameters	Value:1		
	Links	RotationControlOutput		
Transfer	Name	FullRotationWeightsCSign	Type:	ChangeSign
	Description	Change of sign		
	Links	RotationControlOutput		

Table 2. (*continued*)

Transfer	Name	FullRotationWeightsCOutput	Type:	ZeroFilter
	Description	Ensures that the output weight for motor C is non-zero		
	Parameters	Value:1		
	Links	FullRotationWeightsCSign		
Controller		LocationRotationSensorControl		
		InputFunctions		
Input	Name	LocationRotationSensorControlInput	Type:	WeightedMultiply
	Description	The smoothed light sensor value, if the full rotation has finished		
	Parameters	Weights:1,1;Root:false		
	Links	SensorSmooth,LocationRotationReferenceGreaterThan		
		ReferenceFunctions		
Reference	Name	LocationRotationSensorControlReference	Type:	WeightedMultiply
	Description	Reference of light value		
	Parameters	Weights:1,1,1;Root:false		
	Links	LocationRotationReferenceGreaterThan,LocationRotationReferenceWeighted PartialSequenceControlActive		
Transfer	Name	LocationRotationReferenceWeighted	Type:	WeightedMultiply
	Description	The location rotation reference, non-zero provided ControlSwitch is active		
	Parameters	Weights:1,1;Root:false		
	Links	ControlSwitch,ShortMax		
Transfer	Name	LocationRotationReferenceGreaterThan	Type:	GreaterThan
	Description	Test indicating full rotation has finished		
	Parameters	Upper:0;Lower:1		
	Links	SequenceControlOutput,LocationRotationReferenceDiff		

Table 2. (*continued*)

		ErrorFunctions		
Error	Name	LocationRotationSensorControlError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:1;Inactive:0.001		
		OutputFunctions		
Output	Name	LocationRotationSensorControlOutput	Type:	LimitBanded
	Description	Binary indication of goal evaluation		
	Parameters	Threshold:0;BandWidth:1;BandValue:0;Upper:1;Lower:−1		
	Links	LocationRotationSensorControlError		
Transfer	Name	PartialSequenceControlActive	Type:	DigitalLimit
	Description	Indication of 1st two sequence events		
	Parameters	Threshold:0;Upper:0;Lower:1		
	Links	PursuitReferenceGreaterThan		
Transfer	Name	PartialSequenceControlActiveOpposite	Type:	DigitalLimit
	Description	Opposite of previous function		
	Parameters	Threshold:0;Upper:0;Lower:1		
	Links	PartialSequenceControlActive		
Transfer	Name	LocationRotationWeightBSign	Type:	ChangeSign
	Description	Ensures weight signs for each motor will be different resulting in rotation		
	Links	LocationRotationSensorControlOutput		
Transfer	Name	LocationRotationWeightsBOutput	Type:	ZeroFilter
	Description	Converts 0 values of location rotation output ensuring non-zero weight		
	Parameters	Value:1		
	Links	LocationRotationWeightBSign		
Transfer	Name	LocationRotationWeightsCOutput	Type:	ZeroFilter
	Description	Converts 0 values of location rotation output ensuring non-zero weight		

Table 2. (*continued*)

	Parameters	Value:1		
	Links	LocationRotationSensorControlOutput		
Controller		PursuitControl		
		InputFunctions		
Input	Name	PursuitControlInput	Type:	WeightedMultiply
	Description	Smoothed value of light sensor, filtered		
	Parameters	Weights:1,1;Root:false		
	Links	SensorSmooth,PursuitReferenceGreaterThan		
		ReferenceFunctions		
Reference	Name	PursuitControlReference	Type:	WeightedMultiply
	Description	Reference value of light		
	Parameters	Weights:1,1;Root:false		
	Links	PursuitReferenceGreaterThan,PursuitReferenceWeighted		
Transfer	Name	PursuitReferenceGreaterThan	Type:	GreaterThan
	Description	Test indicating full rotation and location has finished		
	Parameters	Upper:0;Lower:1		
	Links	SequenceControlOutput,PursuitReferenceDiff		
Transfer	Name	PursuitReferenceWeighted	Type:	WeightedMultiply
	Description	The pursuit reference, non-zero provided Control Switch is active		
	Parameters	Weights:1,1;Root:false		
	Links	ControlSwitch,LongMax		
		ErrorFunctions		
Error	Name	PursuitControlError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:1;Inactive:0.001		

Table 2. (*continued*)

		OutputFunctions		
Output	Name	PursuitControlOutput	Type:	LimitBanded
	Description	Binary indication of pursuit active		
	Parameters	Threshold:0;BandWidth:1;BandValue:0;Upper:1;Lower:−1		
	Links	PursuitControlError		
Level		3		
Controller		SequenceControl		
		InputFunctions		
Input	Name	SequenceControlInput	Type:	WeightedSum
	Description	The sum of the inputs from the three events in the sequence		
	Parameters	Weights:1,1,1		
	Links	RotationControlLimit,LocationRotationSensorControlInput PursuitControlInput		
Transfer	Name	RotationControlLimit	Type:	DynamicLimit
	Description	The rotation control input limited by the rotation control reference		
	Links	RotationControlInput,RotationControlReference		
		ReferenceFunctions		
Reference	Name	SequenceControlReference	Type:	WeightedSum
	Description	Reference value of sequence		
	Parameters	Weights:1,1,1		
	Links	FullRotationReferenceWeighted,LocationRotationReferenceWeighted PursuitReferenceWeighted		
Transfer	Name	FullRotationReference	Type:	Constant
	Description	A constant reference value		
	Parameters	Constant:750		

Table 2. (*continued*)

Transfer	Name	FullRotationReferenceWeighted	Type:	WeightedMultiply
	Description	The reference value, filtered		
	Parameters	Weights:1,1;Root:false		
	Links	ControlSwitch,FullRotationReference		
Transfer	Name	LocationRotationReferenceDiff	Type:	Subtract
	Description	Difference between sequence control reference and full rotation reference		
	Links	SequenceControlReference,FullRotationReferenceWeighted		
Transfer	Name	PursuitReferenceDiff	Type:	Subtract
	Description	Difference between location rotation diff and location rotation ref		
	Links	LocationRotationReferenceDiff,LocationRotationReferenceWeighted		
		ErrorFunctions		
Error	Name	SequenceControlError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:1;Inactive:0.001		
		OutputFunctions		
Output	Name	SequenceControlOutput	Type:	PassThrough
	Description	Sequence amount remaining		
Controller		SonicControl		
		InputFunctions		
Input	Name	SonicSmooth	Type:	Smooth
	Description	Smoothed value of sonic sensor		
	Parameters	Smoothness:0.25		
	Links	USensorMedian		

Table 2. (*continued*)

		ReferenceFunctions		
Reference	Name	SonicReference	Type:	Constant
	Description	Reference value of sonic sensor		
	Parameters	Constant:30		
		ErrorFunctions		
Error	Name	SonicError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:−1		
		OutputFunctions		
Output	Name	SonicOutput	Type:	Scaling
	Description	Scaled value of error		
	Parameters	Scale:2;Offset:100		
Transfer	Name	SonicControlSpeed	Type:	WeightedMultiply
	Description	Filtered speed as function of error		
	Parameters	Weights:1,1;Root:false		
	Links	SonicOutput,PartialSequenceControlActiveOpposite		
Transfer	Name	SequenceControlSpeed	Type:	WeightedMultiply
	Description	Reference value of speed, filtered		
	Parameters	Weights:1,1;Root:false		
	Links	SpeedConstant,PartialSequenceControlActive		
Transfer	Name	SpeedReference	Type:	WeightedSum
	Description	The sum of speeds from sonic and sequence, only one will be non-zero		
	Parameters	Weights:1,1		
	Links	SequenceControlSpeed,SonicControlSpeed		

Table 2. (*continued*)

Level		4		
Controller		SensorSumChangeControl		
		InputFunctions		
Input	Name	SmoothSensorSumChange	Type:	Smooth
	Description	The smoothed value of the sum of the changes		
	Parameters	Smoothness:0.5		
	Links	SensorSumChanges		
Transfer	Name	MotorBChange	Type:	Change
	Description	Change in B motor speed		
	Parameters	Absolute:true		
	Links	MotorBInput		
Transfer	Name	MotorCChange	Type:	Change
	Description	Change in C motor speed		
	Parameters	Absolute:true		
	Links	MotorCInput		
Transfer	Name	SonicChange	Type:	Change
	Description	The change in the value of the sonic sensor		
	Parameters	Absolute:true		
	Links	SonicSmooth		
Transfer	Name	SensorChangeAbsolute	Type:	Change
	Description	Absolute value of light sensor change		
	Parameters	Absolute:true		
	Links	LightSensorRead		
Transfer	Name	SensorSumChanges	Type:	WeightedSum
	Description	Sum of partial sequence and changes in the motor, sonic, light sensors		
	Parameters	Weights:1,1,1,1,1		

Table 2. (*continued*)

	Links	MotorBChange,MotorCChange		
		SonicChange,SensorChangeAbsolute,PartialSequenceControlActive		
		ReferenceFunctions		
Reference	Name	SensorSumChangeReference	Type:	Constant
	Description	The reference of 0 for the changes		
	Parameters	Constant:0		
		ErrorFunctions		
Error	Name	SensorSumChangeError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:−1		
		OutputFunctions		
Output	Name	SensorSumChangeOutput	Type:	LimitBanded
	Description	Binary indication of error		
	Parameters	Threshold:0;BandWidth:0.1;BandValue:1;Upper:0;Lower:0		
	Links	SensorSumChangeError		
Transfer	Name	SensorSumChangeHabituate	Type:	Habituate
	Description	Habituating value of error		
	Parameters	Rate:0.2;Threshold:0.1;Proportional:false		
	Links	SensorSumChangeOutput		
Transfer	Name	SensorSumChangeReverseWeight	Type:	DigitalLimit
	Description	A value of −1 if SensorSumChangeHabituate is gt 0		
	Parameters	Threshold:0;Upper:−1;Lower:1		
	Links	SensorSumChangeHabituate		
Transfer	Name	SensorSumChangeRotationWeight	Type:	DigitalLimit
	Description	Positive indication of habituation of change sum		

Table 2. (*continued*)

	Parameters	Threshold:0;Upper:2;Lower:1		
	Links	SensorSumChangeHabituate		
Controller		Max		
		InputFunctions		
Input	Name	LongMax	Type:	Maximum
	Description	The maximum value detected by the light sensor		
	Links	LightSensorRead		
		OutputFunctions		
Output	Name	ShortMax	Type:	Maximum
	Description	The maximum value detected by the light sensor, attenuates over time		
	Parameters	AttenuationRate:0.995		
	Links	SensorSmooth		
Level		5		
Controller		SequenceSwitch		
		InputFunctions		
Input	Name	SequenceSwitchInput	Type:	WeightedMultiply
	Description	Indicates if unit has come close to an object		
	Parameters	Initial:1;Weights:1,1;Root:false		
	Links	SonicErrorLimit,SSWeight		
Transfer	Name	SonicErrorLimit	Type:	LimitBanded
	Description	Binary indication of being close to object		
	Parameters	Threshold:0;BandWidth:10;BandValue:0;Upper:1;Lower:1		
	Links	SonicError		
Transfer	Name	SSWeight	Type:	WeightedSum
	Description	Maintains a weight triggered by a pulse		

Table 2. (*continued*)

	Parameters	Weights:1,1		
	Links	SequenceSwitchInput,PartialActiveChangeLimit		
		ReferenceFunctions		
Reference	Name	SequenceSwitchReference	Type:	Constant
	Description	Reference value of indicating not proximal		
	Parameters	Constant:0		
		ErrorFunctions		
Error	Name	SequenceSwitchError	Type:	Subtract
	Description	Difference between goal and current perception		
	Parameters	Sign:1		
		OutputFunctions		
Output	Name	SequenceSwitchOutput	Type:	LimitBanded
	Description	Indication of switching off sequence		
	Parameters	Threshold:0;BandWidth:0.5;BandValue:0;Upper:1;Lower:1		
	Links	SequenceSwitchError		
Transfer	Name	ControlSwitch	Type:	WeightedSum
	Description	Switching sequence on and off		
	Parameters	Weights:1,1		
	Links	SequenceSwitchOutput,SpaceControlOutput		
Controller		SpaceControl		
		InputFunctions		
Input	Name	SpaceControlInput	Type:	WeightedMultiply
	Description	Perception of being in open space		
	Parameters	Initial:1;Weights:1,1;Root:false		
	Links	SonicErrorSumLimit,SCWeight		

Table 2. (*continued*)

Transfer	Name	PartialActiveChange	Type:	Change
	Description	Indication of partial sequence activation		
	Links	PartialSequenceControlActive		
Transfer	Name	PartialActiveChangeLimit	Type:	DigitalLimit
	Description	A pulse when partial sequence turns off		
	Parameters	Threshold:–0.5;Upper:0;Lower:1		
	Links	PartialActiveChange		
Transfer	Name	SonicErrorSmooth	Type:	Smooth
	Description	A smoothed value of the sonic error		
	Parameters	Smoothness:0.5		
	Links	SonicError		
Transfer	Name	SonicErrorSum	Type:	WeightedSum
	Description	At least 1 if the partial sequence turns off or unit is not proximal		
	Parameters	Weights:1,1		
	Links	SonicErrorSmooth,SequenceSwitchInput		
Transfer	Name	SonicErrorSumLimit	Type:	LimitBanded
	Description	A value of 0 if the smoothed sonic error is 0, otherwise 1		
	Parameters	Threshold:0;BandWidth:0.5;BandValue:0;Upper:1;Lower:1		
	Links	SonicErrorSum		
Transfer	Name	SCWeight	Type:	WeightedSum
	Description	Maintains a weight triggered by a pulse		
	Parameters	Weights:1,1		
	Links	SpaceControlInput,PartialActiveChangeLimit		
		ReferenceFunctions		
Reference	Name	SpaceControlReference	Type:	Constant
	Description	Reference value indicating open space		
	Parameters	Constant:1		

Table 2. (*continued*)

		ErrorFunctions		
Error	Name	SpaceControlError	Type:	Subtract
	Description	Error of space indication		
	Parameters	Sign:1		
		OutputFunctions		
Output	Name	SpaceControlOutput	Type:	PassThrough
	Description	Indication of being in open space		
	Links	SpaceControlError		
Controller		Env		
		InputFunctions		
Input	Name	SpeedConstant	Type:	Constant
	Description	Reference value of speed		
	Parameters	Constant:100		