

Growth and Repair: Instantiating a Biologically-Inspired Model of Neuronal Development on the Khepera Robot

T. Elliott¹ and N.R. Shadbolt

Department of Electronics and Computer Science,
University of Southampton,
Highfield, Southampton, SO17 1BJ,
United Kingdom.

Tel.: +44 (0)23 8059 6000; Fax.: +44 (0)23 8059 3313.

Abbreviated Title: Neuronal Development in a Robot.

Abstract

Activity-dependent, developmental synaptic plasticity permits, in part, an animal's nervous system to be constructed in a way that depends on both the animal's experiences and its body's particular morphology. The continuing expression of this plasticity into later life allows constant adaptation of the animal's nervous system to changes in the animal's body as it grows and ages, also allowing some recovery from major injury. We instantiate a previously-studied model of such neuronal development on a robotic platform to establish whether "developmental robotics" can equip a robot with a nervous system tuned to the robot's individual morphology and that allows recovery from damage.

Keywords: neuronal development; synaptic plasticity; activity-dependent competition; Khepera robot; self-repair.

¹To whom correspondence should be addressed; e-mail: te@ecs.soton.ac.uk.

1 Introduction

The ability to grow a nervous system using mechanisms that are influenced by neuronal activity is perhaps one of the greatest discoveries of evolution. From the moment of birth (and before in the case of spontaneous neuronal activity), it allows ongoing developmental processes to wire up the nervous system of an animal in a manner that is strongly affected by the animal's environment [10]. On this view, the boundary between developmental plasticity and adult learning is unclear. Indeed, in higher vertebrates such as mammals, developmental processes can extend far into the post-natal period, with the cortex still undergoing significant anatomical change.

A ubiquitous feature of these processes is activity-dependent, synaptic competition. Such competition is intimately involved in the growth, elaboration and re-arrangement of neurons' axonal arborisations on target structures, i.e., in the maintenance and growth of neurons' trees of axonal branches forming synapses with post-synaptic cells [10]. Neurons that are disadvantaged during these competitive processes, by being less electrically active than other neurons, end up with small, impoverished axonal arbors, while those that are advantaged, by being more electrically active, end up with large, complex axonal arbors. These processes allow the nervous system to adapt in a variety of ways to variations in environmental factors, but also to variations in an animal's own body [9]. For example, the early loss of a sensory system (e.g., an eye or a hand) allows a remapping of the sensory systems on the cortex.

Robots and animals are not so different: both have to exist in the real world, and the problems that they face, and perhaps therefore the solutions that they should find to these problems, are similar. "Developmental robotics" therefore

seems a natural field: the equipping of a robot with a model of neuronal development should enable, in part, the robot's nervous system to grow and be wired up in a way that is conditioned by its environment, the environment being broadly construed as both the external world and the robot's own body. If evolution has discovered that permitting these influences in animals has adaptive value, then perhaps it will be advantageous for a robot so to be constituted. One possible consequence of this for truly autonomous robots is the capacity for what we might term "self-repair". If a robot's sensory systems are damaged, a developmental model may permit some level of recovery. This adaptation will be at a morphological level (changes in the structure of the robot's nervous system), but those changes may also be associated with some degree of behavioural recovery in tasks whose performance depended on the lost or damaged sensory systems.

Anatomical growth and re-arrangement are characteristic of neuronal development, and understanding the mechanisms of activity-dependent competition is an outstanding problem in neuroscience. Despite this, in simulating the later development of the nervous system (typically the visual system) many modellers are content to ignore anatomical plasticity altogether, instead considering only physiological changes in an anatomically-fixed network of connections. In addition, most modellers impose or enforce competition between neurons by using mathematical devices that have little basis in biology. Some modellers have, however, considered anatomical plasticity, and some have also been seeking to illuminate the mechanisms of competition (see, for example, refs. [1, 8]). We too have been seeking to construct biologically-plausible models of developmental processes. We currently have available a class of models of neuronal development in which anatomical re-arrangement and synaptic competition are two faces of the same coin: one depends critically on the other [3-6]. We have built these

models by appealing to experimental data implicating a group of molecular factors (specifically, neurotrophic factors) both in activity–dependent competition and in anatomical re-arrangement [3–6].

In this paper, we seek to apply our biologically–inspired model of neuronal development, previously considered only in simulation, to a robotic platform. We wish to determine whether running a model of development on a robot enables the robot to develop a sensory map of its sensory periphery and a motor map, coupled to the sensory map, that permits the robot to perform a specified task. We further wish to determine whether developmental plasticity can restore some level of performance in the robot’s task–focused behaviour when we manipulate (“damage”) the robot’s sensory systems.

The plan for the remainder of the paper is as follows. In section 2, we describe the robot that we use in our studies, the task that we give it, and the world it inhabits. Then, in section 3 we discuss and derive our biologically–inspired model of anatomical, competitive, activity–dependent plasticity, and describe its application to the robot we use and the task we set it. Section 4 discusses the control mechanisms for the robot that supplement those implicit in the implementation of its developing neuronal network. Our experimental protocols are outlined in section 5. There follows in section 6 a presentation of our main results. Finally, in section 7, we discuss our results and draw any conclusions.

2 The Robot, the Task and the Environment

2.1 The Robot

We use the miniature, mobile Khepera robot (K–Team, Switzerland) as the platform on which to instantiate our neuronal network architecture and plasticity algorithm, described in the next section (see figure 1). The basic Khepera robot

consists of eight infra-red (IR) sensors arranged around the periphery of the robot, numbered according to the conventions described in the Khepera User Manual, as illustrated in figure 2. Each sensor generates a 10-bit integer value ranging from 0 (no stimulation) to 1023 (maximal stimulation). We scale these values into the interval $[0, 1]$ by dividing a sensor's output by 1023. Sensors are rather variable, with a pair of sensors typically having differing average response profiles under identical conditions. This variability can dramatically affect a robot's performance on behavioural tasks, affecting inter-Khepera comparisons.

We use two Khepera robots in this work and find that differences in the robots' sensors are indeed important. In figure 3 we map the sensors' response profiles on each robot by measuring the responses of the sensors to a small tube of white paper (~ 5 mm in diameter, 4 cm long) rotated around each IR sensor at a distance of 15 mm.

The Khepera robot also possesses two independently controllable wheels that may rotate in both directions and that are used to move the robot around its environment.

A Khepera robot may run autonomously on its own batteries for 20 to 30 minutes while executing a program downloaded into its memory. However, we need continuously to monitor the state of the neuronal network being developed by the robot, and the only effective way of achieving this is to control the robot via a serial port connection (which also supplies the power to the robot), with the program running on a computer and saving data to hard disk. Although this is less elegant than a wireless, independent Khepera, the autonomy of the combined computer-Khepera system is maintained. The wires to the Khepera go via a rotating contact (K-Team, Switzerland) so that they do not become tangled as the robot explores its environment.

2.2 The Task

The first task we set each Khepera robot is to develop a mature sensorimotor map, using the biologically-inspired model of competitive, activity-dependent anatomical synaptic plasticity described later, that enables the robot to avoid obstacles. Thus, we imagine that the robot at first possesses an immature, undeveloped (artificial) nervous system. Due to its undeveloped state, the nervous system at first performs poorly at mediating obstacle avoidance, and certain “reflex” behaviours, discussed later, prevent damage to the robot. Instructed by the patterns of activity sweeping through the nervous system, originating from the robot’s IR sensors, the nervous system develops to a mature state. Development takes the form of competitive, activity-dependent interactions between afferent neurons for control of target neurons. Developmental processes lead to the emergence of a topographic representation of the robot’s sensory periphery (the IR sensors) on a set of sensory neurons,² and to the emergence of a motor map reflecting an ipsilateral–contralateral IR sensor distinction. The appropriate development of this sensorimotor map improves a robot’s performance at obstacle avoidance. For calibration purposes, we compare the performance acquired by these developmental processes to that exhibited by a standard Braitenberg algorithm [2].

The second task each Khepera robot is required to perform is to exhibit damage– or deprivation–induced plasticity of the mature sensorimotor mapping. Artificially depriving an IR sensor by setting its output to zero will degrade the obstacle avoidance performance mediated by a developed mapping. Plasticity of the sensorimotor map in response to such deprivation may allow some compen-

²Strictly we should refer to these neurons as second order sensory neurons as they are one synapse removed from the primary sensory neurons (the IR sensors), or we should refer to them as sensory map neurons. For convenience we refer to them as sensory neurons.

sation for the absence of a sensor’s output, perhaps restoring some of the lost performance.

2.3 The Environment

A robot’s environment is a rectangular arena the size of a sheet of A4 paper. The walls of this arena are lined with either plain white paper or alternating black and white striped paper, the stripes being 14 mm wide (this pattern is obtained by using the narrower stripes displayed in the K-Team’s K213 Linear Vision Unit User Manual when printed, one page per side, on white A4 paper). Obstacle avoidance consists only in wall avoidance in our environments: other than the walls, no other obstacles are present. This, along with our use of an unsupervised learning algorithm for learning a behaviour that has a clear sense of “good” and “bad” performance, is justified later.

3 The Model of Plasticity

In this section we first discuss and derive in general terms our model of anatomical, competitive synaptic plasticity without reference to any particular biological system or implementation in simulation or on a robot. We then discuss the application of the model to the development and plasticity of a sensorimotor map in the Khepera robot.

3.1 General Description

Consider a set of pre-synaptic (afferent) neurons labelled by letters such as i and j , and a set of post-synaptic (target) neurons labelled by letters such as x and y . These letters denote the physical locations of the neurons, and are thus vectors; for notational convenience, we leave the vector character implicit. We denote the

number of synapses between afferent cell i and target cell x by s_{xi} . We assume that target cells release a growth-promoting molecular factor (GPMF) (perhaps a neurotrophic factor, although we need not be committed to this particular group of factors) which may diffuse through the target space, and which is taken up by afferent cells. The central assumption of our model is that the average level of GPMF uptake determines the number of synapses in the region of uptake [4]. Denoting GPMF uptake from target cell x by afferent cell i by u_{xi} , and average uptake by \bar{u}_{xi} , our central assumption is simply

$$s_{xi}(t) = \bar{u}_{xi}(t). \quad (1)$$

Numerous models for the average are possible, but we restrict attention to one particular version for reasons discussed elsewhere [4]. We take the time average to be

$$\bar{u}_{xi}(t) = \int_{-\infty}^t dt' \epsilon(t') u_{xi}(t') \exp\left(-\int_{t'}^t dt'' \epsilon(t'')\right), \quad (2)$$

where $\epsilon(t) = 1/\tau(t)$, and $\tau(t)$ is a time-dependent parameter that sets the instantaneous time-scale for the average. In previous work we have taken $\tau(t) = \tau$, independent of time.

Differentiating Eq. (1) with respect to time, we obtain a simple first-order differential equation for s_{xi} :

$$\frac{ds_{xi}}{dt} = \epsilon(t) [u_{xi}(t) - s_{xi}(t)]. \quad (3)$$

This equation encodes our basic requirements: if the current, instantaneous level of GPMF uptake exceeds the current, average level of uptake, then sprout more synapses; if current uptake falls below average uptake, then retract synapses. The overall rate at which this occurs is determined by the time-dependent learning rate $\epsilon(t)$.

A particular model for u_{xi} is derived as follows. First, suppose that target cells produce and release the GPMF in an activity–dependent manner [4]. If the activity of afferent i is denoted by $a_i \in [0, 1]$, then we take the GPMF release from target cell x to be

$$r_x = T_0 + T_1 \frac{\sum_i s_{xi} a_i}{\sum_i s_{xi}}, \quad (4)$$

so that there is an activity–independent component in the release, T_0 , and a bounded activity–dependent component, with a maximum of T_1 . This particular form is the simplest, parameter–free selection satisfying these requirements. This released factor then diffuses through the target field, the diffusion being characterised by the function Δ_{xy} , so that the level at each target cell following diffusion is given by

$$d_x = \sum_y \Delta_{xy} r_y. \quad (5)$$

In writing down this expression we have made the simplifying assumption that the GPMF diffusion from each target cell is independent of the level of GPMF diffusing from other target cells. Relaxing this assumption entails solving the diffusion equations for this system, a complication that we consider to be unwarranted given the relatively coarse characterisation of the model’s dynamics. The diffused GPMF is then taken up by afferents in a manner that we assume to be determined by the number of synapses that an afferent projects into a local volume of tissue; the activity of the afferent; the number of receptors for the GPMF that each of an afferent’s pre–synaptic terminals expresses on its membrane, which we denote by ρ_i . Thus, we assume that

$$u_{xi} \propto s_{xi} g(a_i) \rho_i d_x. \quad (6)$$

We take $g(a_i) = a + a_i$ as a simple model governing the activity–dependence of GPMF uptake by afferents, where the parameter a determines the level of resting

uptake. We assume that the number of receptors per pre-synaptic terminal depends on an afferent's average activity, \bar{a}_i , and is inversely proportional to the afferent's total number of synapses, so that $\rho_i = \bar{a}_i / \sum_x s_{xi}$ [4]. Finally, to fix the constant of proportionality in Eq. (6), we assume that, in any given fixed time interval, the amount of GPMF taken up by afferents matches that supplied by target cells, i.e., $\sum_i u_{xi} = d_x$. Thus, the uptake of GPMF is given by the expression

$$u_{xi} = \frac{s_{xi}(a + a_i)\rho_i}{\sum_j s_{xj}(a + a_j)\rho_j} \sum_y \Delta_{xy} \left(T_0 + T_1 \frac{\sum_j s_{yj}a_j}{\sum_j s_{yj}} \right). \quad (7)$$

This, together with Eq. (3), gives the final form of our model of plasticity:

$$\frac{ds_{xi}}{dt} = \epsilon(t)s_{xi} \left[\frac{(a + a_i)\rho_i}{\sum_j s_{xj}(a + a_j)\rho_j} \sum_y \Delta_{xy} \left(T_0 + T_1 \frac{\sum_j s_{yj}a_j}{\sum_j s_{yj}} \right) - 1 \right]. \quad (8)$$

The overall rate of development of a sensory or motor map is set by the function $\epsilon(t)$ in Eq. (8). In previous work in simulation, we took ϵ to be a constant [3–6]. For work on the Khepera robot, we find it necessary to modify the learning rate function. For a sufficiently large value of (constant) ϵ that permits the rapid development of a sensory map (within 30 to 60 minutes), we find that periods during which the robot travels between obstacles, and thus the receptors receive little or no input, induce deprivation-like responses in the maps. Decreasing ϵ so that these episodes are less significant in terms of overall development is, in principle, acceptable, but slows down development to a rate at which it becomes impractical to run more than a small handful of complete experiments in a day. We are therefore driven to introduce a learning rate function that exploits periods of IR sensor stimulation (proximity to an obstacle) by increasing the rate of plasticity, but that decreases the rate of plasticity when IR sensors are unstimulated (travelling between obstacles). Nevertheless, it is necessary that long-term deprivation of a receptor should induce appropriate changes in the anatomy of the

network. A natural way to encode these requirements is to suppose that the level of activity in an afferent pathway gates the level of plasticity in that pathway. A simple form for $\epsilon(t)$, which must now carry a subscript to denote the afferent to which the learning rate applies, is

$$\epsilon_i(t) = \epsilon(\alpha + \beta a_i), \quad (9)$$

and this is the model for the learning rate that we use in Eq. (8).

3.2 Application to the Khepera Robot

We use the plasticity model to drive the development and ongoing plasticity of a sensorimotor map implemented on the Khepera robot. Input to the sensory map neurons in the model comes from the robot’s eight IR sensors (the “receptors”) around its periphery. These sensory neurons synapse on motor neurons, which in turn drive the Khepera’s two wheel motors (the “muscles”). We discuss the application of the plasticity model to the sensory map and to the motor map separately.

3.2.1 Sensory Map

Because the Khepera’s IR sensors are arranged in a circle, we take the sensory neurons on which the receptors synapse also to be arranged in a circle. The effect of this is to alter neighbourhood relations, and thus distances, between sensory neurons, which affects the initial pattern of connectivity. The minimum number of sensory neurons required faithfully to represent the receptors is eight, and this is what we use. Additional sensory neurons would not confer any advantage in terms of our current aims.

We set up the initial mapping from receptors to sensory neurons as follows. If the topographic map were perfect, each receptor would innervate precisely one

sensory neuron, and immediately adjacent receptors would innervate immediately adjacent sensory neurons. This defines eight different although topologically equivalent patterns of connectivity, but we define the “perfect” mapping as the natural one: the one in which, considering the two circles of receptors and sensory neurons to be super-imposed, a receptor innervates only the sensory neuron in register with it. Let d denote the distance, in cell spacings (bearing in mind the toroidal topology) between any given sensory neuron and the sensory neuron that a particular receptor would innervate were topography perfect. Then, initially, we set the number of synapses between the particular receptor and the given sensory neuron to be proportional to

$$b_t(1 - d/d_{\max}) + (1 - b_t)n, \quad (10)$$

where d_{\max} is the maximum possible distance ($d_{\max} = 4$), $n \in [0, 1]$ is a randomly selected number for each pair of such cells, and $b_t \in [0, 1]$ is a global parameter setting the initial topographic bias in the projections [7]. If $b_t = 0$, then there is no bias at all, so that an afferent cell innervates all target cells randomly. If $b_t = 1$, then the bias is greatest, with no random element in the initial pattern of innervation: each afferent neuron projects synapses in such a way that its topographically perfect target cell receives the greatest number of synapses, and more distant target cells receive fewer synapses, the number falling linearly with distance from the topographically perfect target cell.

For the obstacle avoidance task that the robot is required to perform, it is important that, during development, the mapping from receptors to sensory neurons refines and moves towards the perfect state defined above. However, because the robot has a propensity to move forward (described later), there is a considerable difference between the level of stimulation received by the six front-pointing

receptors and the two rear-pointing receptors. Because the competition between the receptors for control of the sensory neurons depends on activity, the expected, and under normal circumstances desirable, result is that the developed map tends to become distorted, with the front receptors invading the territory that otherwise would have been controlled by the rear receptors. These distortions have a significant impact on the robot’s obstacle avoidance capacity. A straightforward and natural way to prevent these distortions from developing is to introduce a dorsal–ventral distinction in the “neuroanatomy” of the Khepera’s artificial nervous system and remove all synapses from the dorsal (rear) receptors to ventral (front) sensory neurons and from ventral receptors to dorsal sensory neurons. If the number of synapses between an afferent and a target cell is zero, then Eq. (8) does not allow their regrowth.

For the receptor–sensory neuron synapses, we use the following plasticity model parameter settings. The parameters T_0^{SN} and T_1^{SN} (the superscript ‘SN’ denoting sensory neuron) are set, respectively, to 0 and 10. We set $T_0^{\text{SN}} = 0$ because here we are not concerned with demonstrating that our competitive plasticity model can prevent afferent segregation above a critical value of T_0^{SN} [4], and $T_0^{\text{SN}} = 0$ leads to the greatest rate of segregation, i.e., map development. The value of T_1^{SN} is essentially arbitrary, as it sets the overall scale for the numbers of synapses. The value $T_1^{\text{SN}} = 10$ is consistent with values used in previous work [5].³ The parameter a^{IR} (the superscript ‘IR’ denoting IR sensor), which sets the resting uptake of GPMF by IR sensor terminals, is set to $a^{\text{IR}} = 1$; previous work has shown that it must be neither too large nor too small [4]. The diffusion function characterising diffusion of the GPMF through the sensory neuron

³Although synaptic number should naturally be a non–negative integer, for us it is actually continuous–valued. This is unimportant: we can take our synaptic number, scale it and discretise it, as we have done in previous work [5].

field is assumed, for simplicity, to be a Gaussian appropriately normalised on a torus. Its characteristic width, σ^{SN} , is set to $\sigma^{\text{SN}} = 0.75$, again consistent with previous work. Increasing or decreasing σ^{SN} increases or decreases the size of any neuronal assemblies (groups of neurons with similar receptive field properties) that might emerge during development, although no such assemblies could arise in our current work because of our considering so few neurons. The learning rate parameters are set as $\epsilon^{\text{IR}} = 0.05$, $\alpha^{\text{IR}} = 0.01$ and $\beta^{\text{IR}} = 1.00$. The value $\epsilon^{\text{IR}} = 0.05$ is roughly optimal for our purposes: any larger and the sensory maps begin to exhibit too much dependence on temporally recent stimulation; smaller values do not improve the quality of the maps but merely slow down the experiments. The relative size of α^{IR} and β^{IR} is set for convenience: α^{IR} could safely be 10 times larger, although if it were 10 times smaller, it would slow down too much the response to long-term deprivation.

3.2.2 Motor Map

As the Khepera robot has just two wheels we consider just two motor neurons, one for the control of each wheel. Strictly speaking, we could consider four motor neurons, a pair of neurons controlling a wheel, one driving it forwards (agonistic) and the other driving it backwards (antagonistic). However, the two motor neurons that we use drive the wheels backwards against the wheels' propensity to drive the robot forwards at a fixed speed.

In order for the robot to learn a specific task such as obstacle avoidance, it would normally be necessary to employ a supervised or re-inforcement learning algorithm on the sensory to motor neuron mapping, as there is a clear sense of a “required performance”. Because the biological plausibility of models of plasticity is important to us, we would therefore be compelled to construct a

model of re-inforcement learning that we consider to be plausible and consistent with known biological data. However, our goal here is not the construction of such a model. Furthermore, we would prefer, if possible, to apply the same model of plasticity to sensory neuron–motor neuron synapses as to IR receptor–sensory neuron synapses. Additionally, we cannot know, for supervised learning at least, what the motor outputs ought to be during deprivation.

The use of a competitive, activity–dependent plasticity algorithm leads to the formation of maps, the precise structure of which is typically not predictable. To acquire obstacle avoidance behaviour, the final structure of the sensory to motor mapping must, however, be *a priori* predictable, assuming a well–developed, topographic sensory map; that is, the final map must be such that sensory neurons project only to contralateral motor neurons. A biological model that resolves this dilemma is the lateral geniculate nucleus, in which the developmental mechanisms are believed to be activity–dependent and competitive, but in which the final innervation of each lamina by the retinae is predictable. In modelling the development of the retinogeniculate pathway, we have previously assumed that the initial patterns of innervation of each lamina by the retinae is such that some laminae experience a contralateral innervation bias, while others experience an ipsilateral innervation bias. These biases tilt the competition so that, all other factors being equal, the initially dominating inputs to a given lamina eventually win the competition [6].

In the current sensory–motor system, we employ the same strategy. To set up an initial pattern of connectivity between sensory and motor neurons, we first set the numbers of synapses between all eight sensory neurons and both motor neurons to be the same, fixed constant. Then, the presumptive ventral, leftward sensory neurons (i.e., those sensory neurons that would be exclusively innervated

by the ventral, leftward IR sensors were the sensory map topographically perfect) have their synapses to the left (ipsilateral) motor neuron (the neuron that drives the left wheel) multiplied by a factor $(1 - b_c)$, and their synapses to the right (contralateral) motor neuron multiplied by a factor $(1 + b_c)$. The reverse is done for the presumptive ventral, rightward sensory neurons. No such scaling is performed for the dorsal sensory neurons (i.e., those innervated only by the dorsal IR sensors), principally because the dorsal sensory neurons project very few synapses to the motor neurons in the final, developed maps. The parameter b_c determines the contralateral bias, and we set it to a value $b_c = 0.2$, leading to an initial 60%–40% split in contralateral–ipsilateral innervation. A lower value of $b_c = 0.1$ (55%–45%) will lead to largely predictable outcomes, although with occasional sensory neuron mis-projections (i.e., sensory neurons innervating the “wrong” motor neuron in the final state).

The parameters used for motor map development are similar to those used for sensory map development. We set $T_0^{\text{MN}} = 0$, $a^{\text{MN}} = 1$, $\epsilon^{\text{SN}} = 0.05$, $\alpha^{\text{SN}} = 0.01$ and $\beta^{\text{SN}} = 1.00$, all consistent with sensory map values. Unlike sensory neurons, there is no useful notion of distance between the two motor neurons, so we do not consider diffusion of GPMF between the two motor neurons; hence, for the motor system, $\Delta_{xy} = \delta_{xy}$, the Kronecker delta. As indicated for the sensory map, the value of T_1^{SN} sets the overall scale for the number of synapses. In the sensory system, the sensory neurons’ release of GPMF depends on $\sum_i s_{xi} a_i / \sum_i s_{xi}$, as does the sensory neurons’ output to motor neurons, where a_i denotes the sensory neurons’ afferents’ outputs (i.e., IR sensor outputs). Hence, the overall scale is unimportant in the sensory system. This could also be the case in the motor system, making the size of T_1^{MN} an irrelevant scale factor. In this case, the motor neurons’ outputs would follow the same model as for the sensory neurons, so that

they would be in the interval $[0, 1]$, then these could be scaled by some constant factor (a “gain”) to supply drive to the wheels. However, we employ a simpler strategy. The release of GPMF from motor neurons still follows the standard model in Eq. (4), but the direct drive to the wheels from the motor neurons is just $\sum_i s_{xi}a_i$. Thus, the size of T_1^{MN} directly controls the overall scale of this drive. We select a value of $T_1^{\text{MN}} = 120$. A smaller value would not give sufficient drive to the robot to reverse away from an obstacle, while a larger value would supply too much drive and may damage the robot’s motors. The speed of each wheel is set to

$$B - \sum_i s_{xi}a_i, \tag{11}$$

s_{xi} being sensory–motor synapse number between sensory neuron i and motor neuron x , a_i sensory neuron output, and B is a constant. We set $B = 10$, which gives the robot a reasonable although not too fast forward speed in the absence of any IR stimulation. The motor neurons are antagonistic to this forward motion, slowing down and perhaps reversing the robot.

Inevitably, the use of an unsupervised plasticity algorithm on the sensory–motor synapses will lead to maps whose performance at obstacle avoidance may be (although, as we shall see, are not always) inferior to maps developed according to a supervised or reward–based algorithm, or hard–wired Braitenberg–style maps. This does not particularly concern us, because our focus here is principally two-fold: first, to determine whether our biologically–inspired anatomical plasticity algorithm can successfully be transferred from simulation to a real–world robotic platform; second, to determine whether, following deprivation, this plasticity algorithm can restore any significant degree of performance, regardless of the algorithm’s absolute level of performance at obstacle avoidance in the undeprived state. This is also why our environments contain no obstacles other than

the walls: our concern is not with obstacle avoidance *per se*, but with the model underlying the development of a sensorimotor map that happens to mediate obstacle avoidance.

4 Control Strategy

Having described the neuronal architecture of the Khepera, we now describe the overall control system. With perfect sensorimotor mappings and perfect obstacle avoidance, the neuronal network alone would suffice: the robot could be driven by the neuronal network's inputs and outputs only. However, because the undeveloped maps perform poorly, other control mechanisms are required.

The basic Khepera robot lacks bumpers or other devices that could inform us directly of a collision with an obstacle. For an autonomous system, any collision or near-collision has to be inferred purely on the basis of IR sensor outputs. It is possible to determine IR sensor outputs as a function of object proximity and then invert the function to deduce object distance from a set of IR sensor outputs. Such a function would take account, for example, of the rather variable average response profiles of the Khepera's IR sensors. However, it would not take account of the moment-to-moment fluctuations in the outputs, causing the distance estimation to fluctuate. But more important, a developing animal simply does not have access to such a function from the outset of its development. In a sense, the goal of development is precisely the acquisition of this function.

The raw IR sensor outputs alone, then, must be used, without any innate, system-specific "inferential mechanisms" to determine whether a collision or near-collision has occurred. A simple way to achieve this is to suppose, for example, that IR sensor outputs are duplicated in a "pain pathway", and that once stimulation exceeds some pain threshold, a reflex withdrawal from the nox-

ious stimulus occurs. Thus, our criterion for collision determination is simply that

$$\sum_i a_i \geq \theta_c, \quad (12)$$

the a_i 's being IR sensor outputs. If θ_c is set too high, then the threshold will never be reached; if θ_c is set too low, then the robot will effectively avoid stimulation completely. We select a value $\theta_c = 2.0$, so that, for example, two immediately adjacent IR sensors must be maximally activated before reflex withdrawal occurs. This value was selected after extensive tests: it allows ample and full stimulation of the IR sensors while preventing the Khepera from actually hitting a wall and perhaps being damaged.

During reflex withdrawal from a collision or near-collision, the robot's neuronal network output is overridden and the wheel speeds are set to those determined by a Braitenberg algorithm [2] until the IR sensors receive no stimulation ($\sum_i a_i < 0.1$). During withdrawal, the neuronal network's synapses remain plastic for reasons of speeding up development, although we could switch off plasticity during these episodes. Before normal control resumes after a withdrawal event, the robot turns randomly by $\pm 90^\circ$ or 180° .

At the outset, before much development has occurred, the sensorimotor maps are such that the robot occasionally approaches an obstacle, stops and remains stationary without a collision threshold being reached. We therefore employ a "boredom" rule that ensures that if such a situation arises, the robot will turn randomly (by $\pm 90^\circ$ or 180°) and wander off in a different direction. To determine when boredom arises, we calculate a running average of the robot's root mean square (RMS) wheel speed according to the formula

$$\bar{s}_{\text{new}} = \bar{s}_{\text{old}} + (1 - \epsilon_s)(s_{\text{current}} - \bar{s}_{\text{old}}), \quad (13)$$

where \bar{s} is the robot’s average RMS wheel speed and s_{current} is the robot’s current RMS wheel speed, where

$$s_{\text{current}} = \sqrt{s_L^2 + s_R^2}, \quad (14)$$

where s_L and s_R are the speeds of the left and right wheels according to Eq. (11). When $\bar{s}_{\text{new}} \leq \theta_b$, the robot wanders off in a different direction. Empirically, we find that $\epsilon_s = 0.5$ and $\theta_b = 3.0$ give a reasonable boredom threshold: the robot will remain in the same location for approximately one second before wandering away.

Finally, to ensure uniform exploration of the entire environment, after a fixed number of time steps have elapsed and if the robot’s IR sensors are unstimulated ($\sum_i a_i < 0.1$), the robot performs a random turn of $\pm 90^\circ$ or 180° .

These three control strategies of “reflex withdrawal”, “boredom” and “exploration” are all that suffice, in addition to the neuronal network mechanisms themselves, to ensure the robust development of a sensorimotor map that itself can perform obstacle avoidance.

5 Experimental Protocols

We now briefly describe our protocols for performing experiments with the system described above. A Khepera robot, controlled by a DEC alpha computer through one of the computer’s serial ports via a rotating contact, is placed in its environment. A program, running on the alpha, then controls the Khepera, reading the robot’s IR sensors and writing speeds to its wheels, by communicating with the Khepera’s BIOS at a baud rate of 9,600 bps. This program implements all aspects of robot control: the neuronal network, reflex withdrawal, boredom and exploration.

For a given selection of parameters and a particular environment (plain or

striped walls), one complete run consists in the following steps. First the neuronal network's synapses are initialised as described earlier. The program then runs for 10,000 time steps, each time step corresponding to one update of the neuronal network's activity patterns and synaptic connectivity. At each time step, the robot's IR sensors are read once, and the wheel speeds are set once. After each time step, there is a time delay of 50 ms. We find that a delay is necessary to avoid overloading the Khepera, resulting in spurious IR sensor readings and jerky movement. During these first 10,000 time steps, deprivation is not applied to any IR sensor. By around 2,000 time steps, the crash rate (the number of collisions per 1,000 time steps) has reduced and stabilised and the sensorimotor maps are approaching maturity.

At 10,000 time steps, we record the neuronal network's pattern of connectivity for later re-instantiation, then we deprive one or a pair of adjacent IR sensors, by which we mean that, for the purposes of input to the neuronal network's sensory neurons, the deprived IR sensors' outputs are set to zero. When we deprive only one sensor, we deprive receptor number 1, the ventral middle-left IR sensor. This sensor (or its right-hand equivalent) was selected for morphological reasons: the arrangement of the ventral receptors is such that receptor number 1 is flanked most closely by its neighbouring receptors and at angles that might allow this receptor's function to be substituted by its immediate neighbours' functions. When we deprive a pair of receptors, we select receptors numbers 1 and 2 or receptors numbers 2 and 3.

In this deprived state, while continuing to permit plastic changes in the neuronal network's architecture, the program runs for a further 5,000 time steps. Any compensation that occurs in the neuronal network's architecture to the deprivation is completed by about 2,000 time steps.

After these 5,000 time steps have elapsed, the neuronal network’s architecture is restored to the state that obtained immediately prior to the onset of deprivation, at 10,000 time steps, and the time counter is also reset to 10,000. The program then continues to run, with the same IR sensor(s) deprived, but not permitting any plasticity in the neuronal network, i.e., the network’s synapses are frozen. In this implastic phase, the program runs for a final 5,000 time steps.

Thus, one run consists of 20,000 time steps, each with a time delay of 50 ms, or a little under 17 minutes. This does not include time spent in reflex withdrawal or in the “bored” state, nor real processing time. In fact, each run takes around 50 minutes to complete.

For each selection of parameters and environment, at least five runs are performed. The data from these runs — numbers of collisions every 100 time steps — are polled, and the average and standard deviation every 100 time steps is determined. The crash rates given below are determined from these averaged data by least-squares linear fitting methods over 3,000, 5,000 or 8,000 time steps, depending on the data being analysed: from time step 2,000 to 10,000 for the initial phase of development without deprivation; from time step 12,000 to 15,000 during the deprived but plastic phase; from time step 10,000 to 15,000 during the deprived and frozen phase.

For calibration purposes, we run a Braitenberg-style algorithm [2] on both our Khepera robots in each of the two environments that we consider. To ensure a valid comparison, we use the same program described above so that all conditions are identical. The only difference is that we pre-load into the neuronal network a pattern of connectivity giving Braitenberg-style obstacle avoidance. To ensure identical timing, the program performs all the computations, etc., that are required to update the network’s synapses at each time step, but we simply

	Sensory Neuron Number							
	0	1	2	3	4	5	6	7
Left Motor Neuron	0	0	0	16	11	7	0	0
Right Motor Neuron	7	11	16	0	0	0	0	0

Table 1: Synaptic connectivity between sensory neurons and motor neurons to achieve Braitenberg-style obstacle avoidance for calibration purposes.

disable the code that actually performs the update. The connectivity to achieve Braitenberg-style obstacle avoidance, with a base forward speed of $B = 10$, is as follows. The IR sensor to sensory neuron synapses are set to achieve a perfect pattern of topography, so that an IR sensor innervates (with $s_{xi} = 1$) only its topographically perfect sensory neuron and no others. This means that the sensory neurons' outputs exactly duplicate their inputs. The sensory neuron to motor neuron synapses are set as in table 1, where we use an identical labelling convention for the sensory neurons as for the IR sensors. The ventral sensory neurons innervate, with appropriate numbers of synapses, the contralateral but not the ipsilateral motor neurons. There are no synapses from the dorsal sensory neurons to the motor neurons. To obtain crash rates for this Braitenberg-style calibration system, we run the program five times each over 15,000 time steps, averaging the crash rates and performing least-squares linear fitting on the resulting data.

6 Results

We first calibrate the robots' performances with the Braitenberg-style obstacle avoidance algorithm [2]. In table 2 we show the crash rates (number of crashes per 1,000 time steps) of both Kheperas in both environments, where a crash is defined as total IR stimulation exceeding $\theta_c = 2$, as discussed above. We see a marked difference between the two robots. Khepera number 1 performs almost perfectly,

with a crash rate of exactly zero in the plain environment and close to zero in the striped environment. Khepera number 2, however, exhibits a large crash rate in both environments. The crash rates for this robot in both environments do not differ at a statistically significant level, by which we mean that the average crash rates at each time step in both environments do not differ at the one standard deviation (1σ) level. The difference in performance between these two Kheperas is purely a function of their IR sensors. As demonstrated in figure 3, Khepera number 1 possesses generally more acute forward-pointing IR sensors (numbers 2 and 3) than Khepera number 2. It is the poorer performance of these IR sensors in Khepera number 2 that mainly contributes to its much higher crash rate. Increasing the crash threshold θ_c would, of course, decrease the crash rate for Khepera number 2, but with the associated risk that the robot could actually make physical contact with a wall and perhaps be damaged. Conversely, decreasing θ_c for Khepera number 1 would increase its crash rate. While it is entirely possible that different robots could use different crash thresholds, there is no reason *a priori* to expect a robot with more acute sensors (at least initially) to have a lower threshold than a robot with less acute sensors: to assume such an anti-correlation would be effectively to supply the robot with *a priori* knowledge about its own sensory systems, which knowledge it must, presumably, acquire through learning about its own sensory performance.⁴ For this reason, we prefer to keep θ_c identical in both robots.

Having calibrated the two robots' performances with a standard algorithm, we now turn to a discussion of the development of the sensorimotor maps during the first 10,000 time steps when the robots' sensors are not deprived. A typical

⁴It is possible that a robot with a crash rate that it discovers to be high (low) could raise (lower) its crash threshold to compensate, but this is a complication that we do not consider here.

	Plain	Striped
Khepera 1	0.0	0.1
Khepera 2	43.3	43.2

Table 2: Crash rates for the two Khepera robots in the plain and striped environments running a Braitenberg-style obstacle avoidance algorithm.

example of the initial sensorimotor map (for $b_t = 0.5$) is shown in figure 4. We show only the ventral IR sensors (the line of six black circles at the top of the figure, arranged in order from sensor number 0 on the left to sensor number 5 on the right) and the ventral sensory neurons (the line of six grey circles, again arranged in numerical order). The dorsal IR sensors and sensory neurons are not shown, as these are of little interest to us here. The two motor neurons are represented as the two open circles at the bottom of the figure. For later clarity, the left motor neuron (controlling the left wheel) is shown on the right of the figure, while the right motor neuron is shown on the left. The thickness of a line between two neurons denotes the number of synapses from one neuron to the other, a key being shown in figure 5.

In figure 6 we show the sensorimotor map developed by Khepera number 1 in the striped environment after 10,000 time steps, for $b_t = 0.5$. In terms of the sensory map, we see that the initial pattern of connectivity has undergone competitive change: the pattern of connectivity is now broadly “correct”, with an IR sensor innervating its appropriate (in terms of the perfect map) sensory neuron. There are additional connections of restricted horizontal range, in which an IR sensor projects synapses to other, but topographically nearby sensory neurons. The presence of these aberrant connections reflects both the correlations in the activities between adjacent IR sensors, and the diffusion of GPMF between sensory neurons. The motor map has also undergone competitive refinement,

so that sensory neurons finally innervate only their contralateral motor neuron. (Hence our reversing left and right for the motor neurons in these figures, so that the contralateral projections from sensory neurons to motor neurons in the developed maps do not cross and clutter the figures.) There is a small, ipsilateral motor connection in this map that would be eliminated with further development. For this particular robot, this sensorimotor map is quite representative for both environments and for all values of the topographic bias parameter b_t used here ($b_t = 0.25$, $b_t = 0.50$, $b_t = 0.75$ and $b_t = 1.00$).

The equivalent map for Khepera number 2 to that shown in figure 6 for Khepera number 1 is shown in figure 7. Although this map has clearly undergone significant change and some topographic refinement, there are also some mis-projections present. For example, IR sensor number 2 innervates only the contralateral sensory neurons, and projects no synapses at all to its topographically appropriate sensory neuron (number 2). This map is again broadly representative for this robot in the striped environment for different values of b_t . However, many of the maps generated in the plain environment exhibit systematic distortions for $b_t = 0.25$, $b_t = 0.50$ and $b_t = 0.75$, but not for $b_t = 1.00$, for which value of b_t there is no randomness in the initial map and the topographic bias is greatest. Of the 15 maps that we generated for $b_t = 0.25$, $b_t = 0.50$ and $b_t = 0.75$ for Khepera number 2 in the plain environment (five for each value of b_t), ten of them exhibit distortions. The distortions fall into two classes, illustrated in figure 8. In the first class, IR sensor number 0 systematically mis-projects contralaterally; in the second class, sensory neuron number 0, now correctly receiving input from IR sensor number 0, systematically mis-projects contralaterally. These distortions in the plain environment are likely the result of the generally better left-side performance of Khepera number 2's IR sensors compared to the performance of

its right-side IR sensors, as revealed in figure 3. In the striped environment, this difference will tend to be less important, principally because of the weakening of the correlations in activation between IR sensors numbers 0 and 1.

Following deprivation of one or more IR sensors at time step 10,000, the synapses from the deprived sensors are slowly removed while those from neighbouring sensors grow and invade the territory vacated by the deprived sensors. This is revealed in figures 9 to 12. All these maps are generated with $b_t = 0.5$ in the striped environment; the data are qualitatively similar for other values of b_t and in the plain environment. Figures 9 to 11 show the impact on the sensorimotor mappings developed by Khepera number 1 of depriving IR sensor number 1, IR sensors numbers 1 and 2, and IR sensors numbers 2 and 3, respectively. Figure 12 shows deprivation of IR sensor number 1 for Khepera number 2. We show both the state of the sensorimotor maps immediately prior to deprivation, at time step 10,000, and after deprivation, at time step 15,000. Following deprivation, we see that deprived IR sensors retract all their synapses to the sensory neurons and that neighbouring IR sensors compensate by increasing their synapses on the sensory neurons that have lost input from deprived sensors.

Having discussed qualitatively the development and plasticity of sensorimotor maps by our two Khepera robots, we can now discuss quantitatively the robots' performance at obstacle avoidance during the three states that our robots experience: undeprived and plastic; deprived and plastic; deprived and implastic. In figures 13 and 14 we plot, for $b_t = 0.25$ in the striped environment, averaged crash rates for Kheperas numbers 1 and 2, respectively, in all three phases, where we deprive IR sensor number 1 in each robot. The error bars denote one standard deviation at each time step. Least-squares fitting the undeprived average performance in the time step range 2,000–10,000 to a straight line, the crash rate

of Khepera number 1 is 10.2 while that of Khepera number 2 is 13.6. Following deprivation but retaining plasticity, the crash rates are 11.1 and 33.2 (fitted over time steps 12,000–15,000), respectively; without plasticity the crash rates are 18.6 and 59.6 (fitted over time steps 10,000–15,000).

We see that, during the undeprived phase, the crash rates for both robots decrease until they reach a roughly constant rate, although the crash rates never reach zero. The development of the sensorimotor maps is largely complete by around 2,000 time steps, and it is during this initial period that the crash rates decrease and finally stabilise. (The initial phase of zero crashes for Khepera number 1 is caused by the robot approaching objects and stopping before a crash occurs; the “boredom” rule then takes over and moves the robot away.) The crash rate of 10.2 for (undeprived) Khepera number 1 is, as expected, larger than the nearly zero crash rate for the same robot in the same environment calibrated on the Braitenberg-style algorithm. However, for Khepera number 2, the crash rate of 13.6 is lower than the crash rate of 43.2 for the Braitenberg algorithm. Furthermore, over the first 10,000 time steps, the Braitenberg algorithm’s performance and that of the (averaged) learned map differ significantly after about 2,000 time steps at the 1σ level. (Indeed, the difference is so large at 10,000 time steps that the significance exceeds the 10σ level.) Although the two robots’ crash rates during the undeprived phase are similar, the performances differ significantly after about 2,500 time steps, with Khepera number 1 doing better than Khepera number 2. Thus, although Khepera number 1 does not perform as well as its Braitenberg capacity, it is still performing pretty well relative to the other robot.

Following deprivation of IR sensor number 1, the crash rates for both robots increase. Khepera robot number 1’s crash rate in the implastic phase is nearly

double the robot’s undeprived performance, but plasticity allows adaptation to the deprivation, reducing the crash rate to nearly its initial value. However, although the plastic system does better than the implastic system in terms of the absolute level of crash rates, the two systems’ performances do not differ at the 1σ level over the 5,000 time steps during which deprivation is applied. It may be possible that such a difference will emerge by running the experiments for much longer, but any significant recovery from deprivation should be fairly rapid (at least as fast as it takes to develop the original, pre–deprivation sensorimotor map) for it to be of any adaptive advantage, and to be of selective potential during evolution. In contrast, while Khepera robot number 2’s deprived performance is much worse than the other robot’s, the crash rate following recovery from deprivation does differ significantly, after a short time, from the implastic system’s performance. Furthermore, although deprived, the plastic system’s performance is still better than the undeprived Braitenberg algorithm.

Tables 3 and 4 summarise our data on crash rates in both robots in both environments, and for all values of b_t used. There are several observations. First, for Khepera number 1, although the robot always performs better at obstacle avoidance following deprivation with plasticity than without plasticity, the difference never becomes statistically significant. Although there is noise in the data, we can discern a very general trend in these data, in that smaller values of b_t tend to produce larger crash rates than larger values. This is particularly noticeable by comparing data for $b_t = 0.25$ to data for all other values. This is to be expected, as topographic sensory map formation will be more reliable with larger values of b_t . Second, for Khepera number 2, the striped environment is always associated with a statistically significant recovery of performance following deprivation. Furthermore, post–recovery performance is always better

b_t	Plain				Striped			
	Runs	\neg D	D \neg P	DP	Runs	\neg D	D \neg P	DP
0.25	5	9.3	15.1	10.5	8	10.2	18.6	11.1
0.50	5	6.9	8.6	5.4	5	5.6	13.6	9.0
0.75	5	4.7	7.4	5.5	5	6.5	13.8	6.9
1.00	5	8.4	10.4	6.7	5	6.9	11.4	9.7

Table 3: Crash rates for Khepera robot number 1 in the two environments. The first column indicates the value of b_t , the topographic bias, used in establishing the initial connections; the third and seventh show the crash rates for both environments during the non-deprived phase of learning (“ \neg D”); the fourth and eighth the crash rates following deprivation but with plasticity switched off (“D \neg P”); the fifth and ninth the crash rates following deprivation but with continued plasticity (“DP”). The second and sixth columns indicate the number of runs considered, a “ \star ” next to the number indicating a statistically significant difference (at the 1σ level) between the D \neg P and the DP data after some number of time steps. All in cases, IR sensor number 1 is deprived at time step 10,000.

than undeprived-Braitenberg performance. Third, for $b_t = 1.00$ in the plain environment, we again observe a significant improvement in post-deprivation performance compared to an implastic system for Khepera number 2. For this value of $b_t = 1.00$, this robot’s performance is better in this environment than for any other value of b_t . For other values of b_t , as discussed above, distortions in the sensorimotor maps develop. These distortions lead to large variances in the data that tend to temper the statistical significance of any post-deprivation performance, although for $b_t = 0.5$ the significance remains.

A significant response to single-receptor deprivation can be robustly observed in Khepera robot number 2, but no significant improvement can be seen in similar manipulations of Khepera robot number 1. With these data alone, therefore, it could be argued that Khepera number 2’s capacity significantly to recover is purely a reflection of its rather poor, general level of performance, compared to robot number 1. Alternatively, the view could be taken that, because the

	Plain				Striped			
b_t	Runs	\neg D	D \neg P	DP	Runs	\neg D	D \neg P	DP
0.25	5	20.0	65.0	44.6	5 *	13.6	59.6	33.2
0.50	5 *	15.2	68.9	38.3	5 *	14.7	62.1	37.1
0.75	5	19.5	62.5	47.4	10 *	13.7	67.0	32.1
1.00	5 *	11.5	66.5	31.2	5 *	15.8	59.3	36.3

Table 4: Crash rates for Khepera robot number 2. The format of this table is identical to that for table 3.

sensorimotor maps formed by Khepera robot number 1 are so good and its two front-pointing IR sensors are somewhat more acute than those of robot number 2, single-receptor deprivation would not be expected to have a significant impact on its performance anyway. To distinguish between these two competing views, we also perform double-receptor deprivation in Khepera robot number 1. All protocols are identical to previous experiments, except that we deprive an adjacent pair of IR sensors in Khepera number 1, IR sensors numbers 1 and 2, or numbers 2 and 3. The data are summarised in table 5. Simultaneously depriving receptors 1 and 2 leads to results similar to those discussed above for this robot, with no statistical significance.⁵ However, depriving receptors 2 and 3, which eliminates input from both forward-pointing receptors, does produce a significant response. The deprived, implastic system crash rate is dramatically enhanced, and although there is recovery with plasticity, post-recovery performance does not approach pre-deprivation performance, as it has in all other forms of deprivation considered here for this robot. Nevertheless, the post-recovery level of performance is impressive: even with its front two IR sensors inactive, this robot’s performance is close to that of Khepera number 2 without any deprivation at all. In figure 15,

⁵The data in the striped environment are on the margins of 1σ statistical significance. In an attempt to reduce noise, we quadrupled the number of runs usually considered.

Pair	Plain				Striped			
	Runs	\neg D	D \neg P	DP	Runs	\neg D	D \neg P	DP
1-2	10	7.6	14.1	6.7	20	6.9	13.0	9.2
2-3	5 ★	7.4	21.9	13.4	5 ★	6.3	33.5	16.0

Table 5: Crash rates for Khepera Robot Number 1. Instead of depriving just one receptor (receptor number 1), as in tables 3 and 4, we deprive a pair of immediately adjacent receptors, the numbers of which are given in column 1. The format of this table is otherwise identical to that for table 3. All data in this table are generated with $b_i = 0.50$.

we show the averaged performance of Khepera number 1 when doubly deprived (receptors 2 and 3) in the striped environment.

7 Discussion

In this paper, we have applied our previously-constructed model of anatomical, activity-dependent, developmental synaptic plasticity to a real-world problem, that of developing a sensorimotor map in the Khepera robot. We have found, barring environmental influences in one of our robots in certain parameter regimes, that sensory map topography emerges robustly, with one robot developing rather precise topographic representations of its sensory periphery and the other less precise but nonetheless clearly topographically-refined sensory maps. By appealing to a tilting of competitive dynamics in favour of certain classes of inputs (the presumptive contralateral sensory neuron inputs), as seems to occur during the development of at least the lateral geniculate nucleus in real nervous systems, we have found that a motor map, whose functional integrity depends critically on the topographically-refined sensory map, also develops robustly. The combined sensorimotor map mediates the avoidance of obstacles (walls) in the robots' environments. One of our robots (Khepera number 2), employing self-

grown sensorimotor maps, performs significantly better than a standard Braitenberg algorithm at obstacle avoidance, with crash rates 2 to 4 times lower than Braitenberg performance. The other robot (Khepera number 1), with somewhat more acute forward-pointing IR sensors, using self-grown maps, performs better than its twin, but worse than its almost perfect Braitenberg performance. It is difficult to perform a meaningful statistical comparison of Khepera number 1's self-grown map performance with its Braitenberg performance, simply because the variance in its Braitenberg performance is virtually zero. However, given that the average crash rate for this robot using its own sensorimotor map can be as low as 4.7 (about 3 times smaller than the other robot's), our overall impression is that Khepera number 1's performance is, in fact, pretty good. We have also shown that deprivation of IR sensors affects dramatically the robots' performance, increasing the crash rate between 2-fold and 5-fold, depending on the robot, the environment, and the topographic bias parameter, b_t . Khepera number 2 shows robust, statistically significant degrees of recovery from single-receptor deprivation, caused by deprivation-induced anatomical re-arrangements in its nervous system. The improvement occurs because the re-arranged nervous system essentially interpolates the deprived IR sensor's output from the outputs of this sensor's two nearest neighbours. Khepera number 1 also shows improvement in performance following single-receptor deprivation, but not at a statistically-significant level. However, double-receptor deprivation can induce a statistically-significant recovery of performance, with this robot's performance even with its front two receptors deprived being as good as the other robot's performance without any deprivation at all.

Our results show some dependence on the type of world inhabited by the robots, whether plain or striped. Although Khepera number 1's undeprived crash

rates in both environments are comparable, when deprived its performance in the plain environment is better than in the striped environment. This is more marked during double-deprivation than during single-deprivation. The difference between the two environments is a simple consequence of the spacing of the stripes in the striped world (14 mm) and the average inter-ventral-receptor spacing (~ 13 mm for both our robots). Thus, in the striped world, when a ventral IR sensor is “seeing” mainly a white stripe, its immediate neighbours are “seeing” mainly black stripes. During single-deprivation, therefore, when the deprived IR sensor is near a white stripe, its neighbouring IR sensors are near black stripes, reducing IR sensor outputs through these channels, and so reducing the re-arranged nervous system’s effectiveness in interpolating the deprived sensor’s output from its neighbours’ outputs. However, when the singly-deprived system’s deprived sensor is near a black stripe, no such problem arises, because its neighbours are near white stripes. During double-deprivation, one of the two immediate, active neighbours of the two deprived IR sensors is always “seeing” a black stripe when the robot is near a wall, therefore always interfering with the nervous system’s attempts at interpolation. For Khepera robot number 2, environmental influences are even greater, inducing systematic distortions in the sensorimotor maps for $b_t \neq 1$ in the plain environment. The result is overall better performance in the striped environment rather than the plain environment for $b_t \neq 1$.

The difference in performance between the two robots, even with a hard-wired Braitenberg algorithm, is very dramatic. During Braitenberg calibration, Khepera number 1 crashed a total of 8 times in 150,000 time steps, all crashes being in the striped world, whereas Khepera number 2 crashed nearly 6,500 times under identical conditions. We are unable to assess whether Khepera number 1’s

performance is untypically good, or whether Khepera number 2's performance is untypically bad, or whether, indeed, they both fall within one standard deviation of the Khepera population's mean level of performance. Whatever the answer, the lesson for robotics is clear: acceptable performance of a given task using a given control solution in one robot, even when that robot's performance is averaged over many runs, does not guarantee the general utility of that control solution.

Why does Khepera robot 2 with a self-grown sensorimotor map perform much better than a hard-wired Braitenberg algorithm, even when deprived of input from one IR sensor or when using a distorted map? One explanation for the importance of developmental mechanisms depending on competitive synaptic growth and re-arrangement is that they permit a nervous system to tune itself to the body in which it finds itself and to the environment in which the body resides [9, 10]. Furthermore, if these "developmental" plasticity mechanisms in fact extend throughout an animal's typical life span, then they allow the nervous system to adapt to changes in the animal's body that occur throughout life. These changes can be the natural wear-and-tear of everyday life, the explosive growth of the body that occurs during puberty, the slow decline of old age, or changes induced by major insults and injury. Our model of anatomical, competitive synaptic plasticity attempts to capture, albeit coarsely, some of the biological principles underlying these mechanisms. A natural answer to our question, therefore, is that our model tunes the developing nervous system of Khepera number 2 to its particular morphological idiosyncrasies in a way in which a hard-wired Braitenberg map obviously does not. As the two front-pointing IR sensors (numbers 2 and 3) on this robot are rather myopic compared to the others, their average level of activity will be lower than normal. Hence, activity-dependent competition will ensure that these sensors are synaptically under-represented in

the sensory neuron layer, with compensation by neighbouring IR receptors. It is this compensation that ultimately provides greater drive to the robot's wheels by permitting adjacent IR sensors to enhance neuronal responses that would otherwise be rather weak.

In overall conclusion, we have shown that a biologically-inspired model of neuronal development can be successfully transferred from computer simulation to robotic instantiation, allowing a robot to “grow” a nervous system adapted to its own body. We have also shown that the same model permits significant recovery from sensory perturbations such as IR sensor deprivation. This capacity for “self-repair” of a “self-grown” nervous system could be useful to robots in dangerous environments in which human intervention is impractical or impossible. More broadly, although we have considered only a simple Khepera robot inhabiting a simple world performing a simple task while running a very coarse and incomplete model of neuronal development, we hope that we have shown that “developmental robotics” is a worthwhile avenue of enquiry.

Acknowledgements: TE thanks the Royal Society for the support of a Royal Society University Research Fellowship.

References

- [1] E.L. Bienenstock, L.N. Cooper, and P.W. Munro. Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* 2:32–48, 1982.
- [2] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. M.I.T. Press, Cambridge, Ma., 1984.
- [3] T. Elliott, A.C. Maddison, and N.R. Shadbolt. Competitive anatomical and physiological plasticity: A neurotrophic bridge. *Biological Cybernetics* 84:13–22, 2001.
- [4] T. Elliott and N.R. Shadbolt. Competition for neurotrophic factors: Mathematical analysis. *Neural Computation* 10:1939–1981, 1998.
- [5] T. Elliott and N.R. Shadbolt. Competition for neurotrophic factors: Ocular dominance columns. *Journal of Neuroscience* 18:5850–5858, 1998.
- [6] T. Elliott and N.R. Shadbolt. A neurotrophic model of the development of the retinogeniculocortical pathway induced by spontaneous retinal waves. *Journal of Neuroscience* 19:7951–7970, 1999.
- [7] G.J. Goodhill. Topography and ocular dominance: a model exploring positive correlations. *Biological Cybernetics* 69:109–118, 1993.
- [8] P.R. Montague, J.A. Gally, and G.M. Edelman. Spatial signaling in the development and function of neural connections. *Cerebral Cortex* 1:199–220, 1991.
- [9] D. Purves. *Body and Brain: A Trophic Theory of Neural Connections*. Harvard University Press, Cambridge, Ma., 1988.

- [10] D. Purves. *Neural Activity and the Growth of the Brain*. Cambridge University Press, Cambridge, 1994.

Figure Captions

Figure 1: The K-Team's miniature, mobile Khepera robot.

Figure 2: Numbering of the Khepera's eight IR sensors on a schematic representation of the robot, using the conventions of the K-Team's Khepera User Manual.

Figure 3: The IR sensors' response profiles for both Khepera robots used here. The inner circle schematically represents a robot's periphery, with each of the small blobs denoting an IR sensor. Seven responses, indicated by the lines, are measured for each IR sensor at angles varying between -30° and $+30^\circ$, inclusive, in steps of 10° , with the direction of the sensor taken as 0° . The length of a line represents the size of the response. The small arcs around each IR sensor denote maximum possible response.

Figure 4: A typical example, for $b_t = 0.5$, of the initial pattern of connectivity between the ventral IR sensors (black circles), the presumptive ventral sensory neurons (grey circles) and the two motor neurons (white circles). The thickness of a line indicates the number of synapses; a key is given in figure 5. For later clarity, the motor neuron that controls the left wheel is shown on the right and the motor neuron that controls right wheel is shown on the left. Synaptic numbers are rounded up or down so that they change in steps of 0.25.

Figure 5: The key for figures such as figure 4, giving the mapping between line thickness and number of synapses. For clarity only integral line thicknesses are shown, although we actually use line thicknesses that increase in steps of 0.25.

Figure 6: A sensorimotor map developed by Khepera number 1 in the striped environment, for $b_t = 0.5$, after 10,000 time steps.

Figure 7: A sensorimotor map developed by Khepera number 2 in the striped environment, for $b_t = 0.5$, after 10,000 time steps.

Figure 8: Typical examples of the two types of distorted sensorimotor maps developed by Khepera number 2 in the plain environment for values of $b_t \neq 1$.

Figure 9: An example of the change in the sensorimotor map induced by depriving IR sensor number 1 in Khepera number 1.

Figure 10: An example of the change in the sensorimotor map induced by simultaneously depriving IR sensors numbers 1 and 2 in Khepera number 1.

Figure 11: An example of the change in the sensorimotor map induced by simultaneously depriving IR sensors numbers 2 and 3 in Khepera number 1.

Figure 12: An example of the change in the sensorimotor map induced by depriving IR sensor number 1 in Khepera number 2.

Figure 13: Number of crashes against time step for Khepera number 1, for $b_t = 0.25$ in the striped environment. For time steps 0–10,000, the robot develops a sensorimotor map without deprivation of any IR sensors (green data). Following deprivation of receptor number 1, the robot's performance at obstacle

avoidance both with plasticity (blue data) and without plasticity (red data) is assessed over time steps 10,001–15,000. Error bars denote one standard deviation.

Figure 14: Number of crashes against time step for Khepera number 2, for $b_t = 0.25$ in the striped environment. The format of this figure is otherwise identical to that of figure 13.

Figure 15: Number of crashes against time step for Khepera number 1, for $b_t = 0.50$ in the striped environment, with IR sensors numbers 2 and 3 simultaneously deprived after time step 10,000. The format of this figure is otherwise identical to that of figure 13.

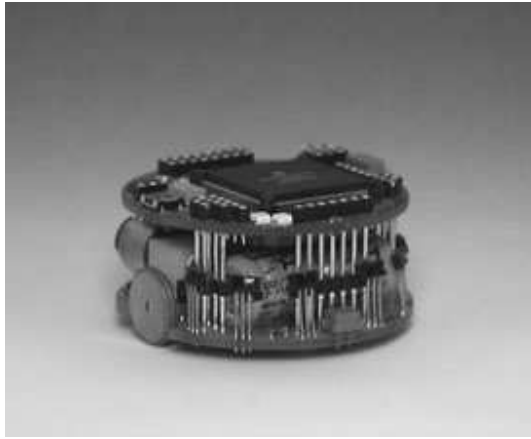


Figure 1.

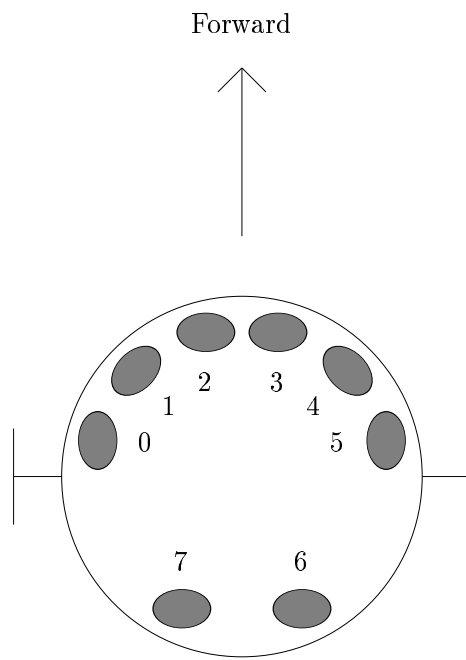
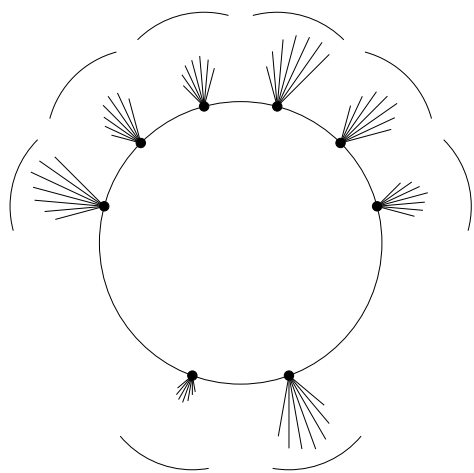
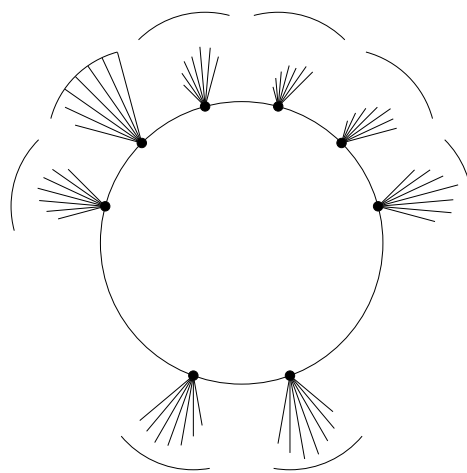


Figure 2.



Khepera number 1



Khepera number 2

Figure 3.

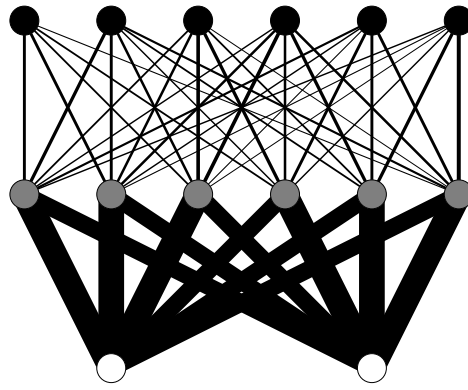


Figure 4.



Figure 5.

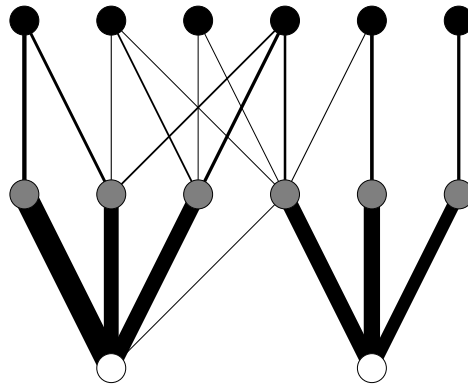


Figure 6.

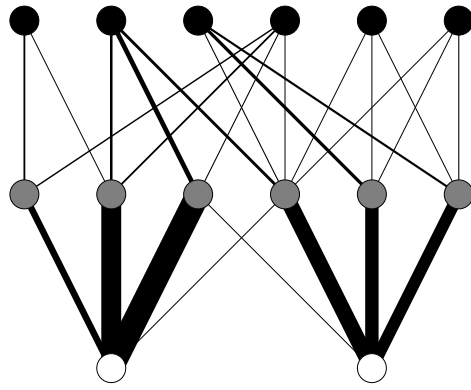


Figure 7.

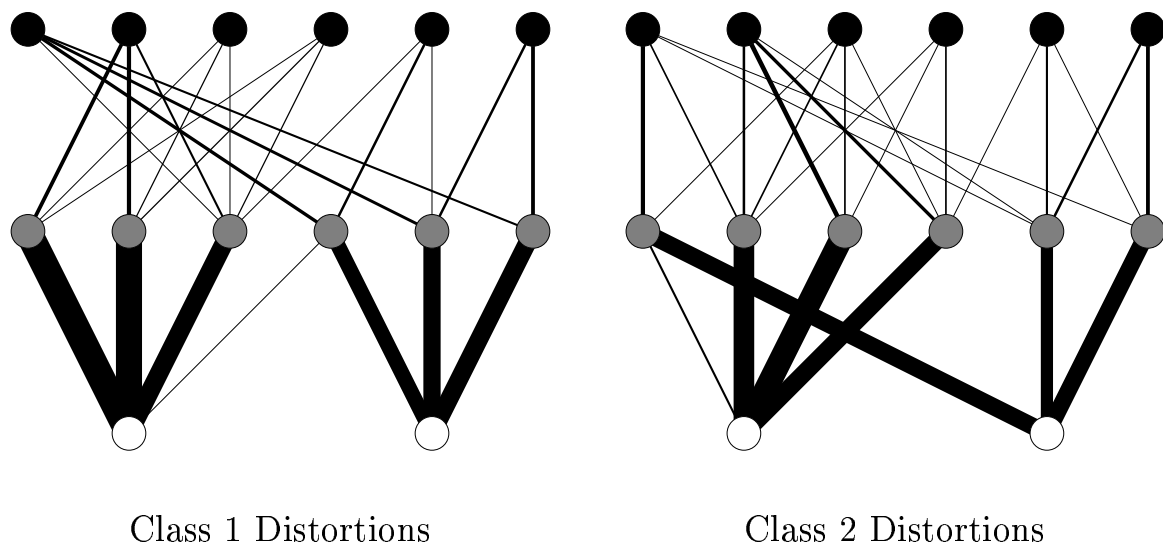


Figure 8.

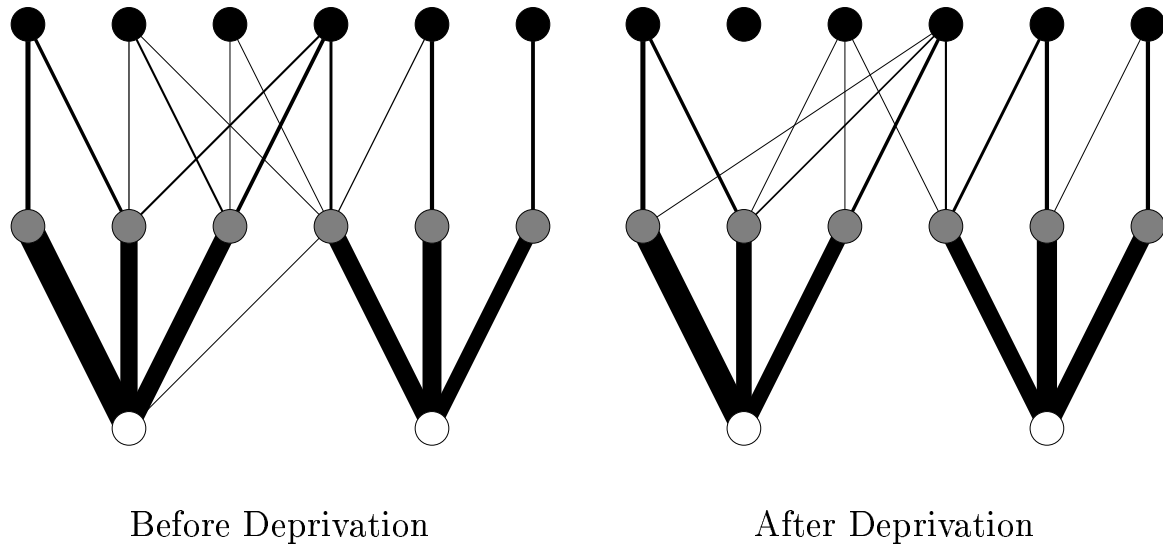


Figure 9.

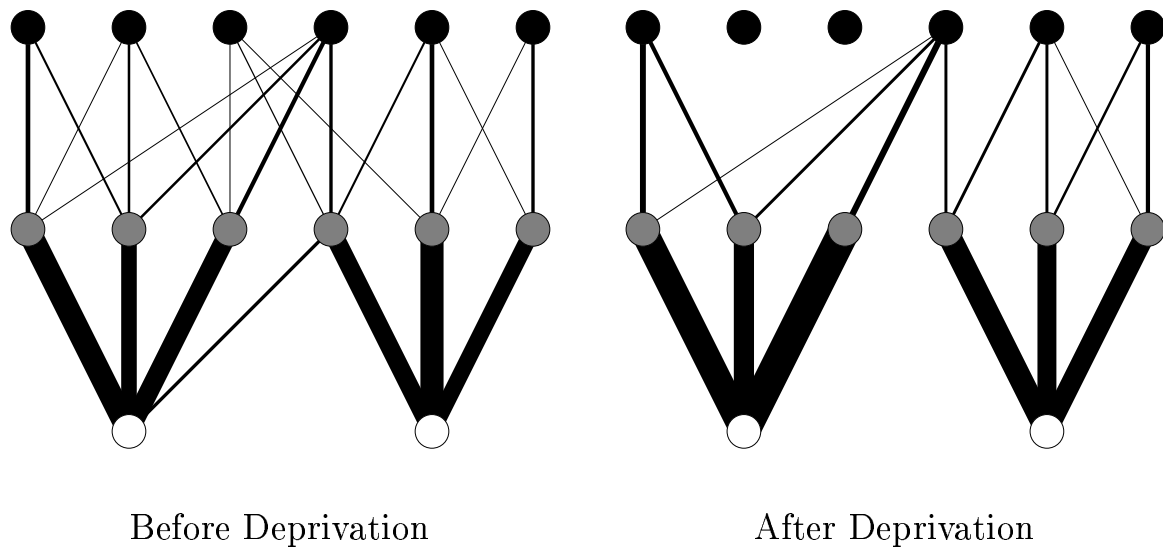


Figure 10.

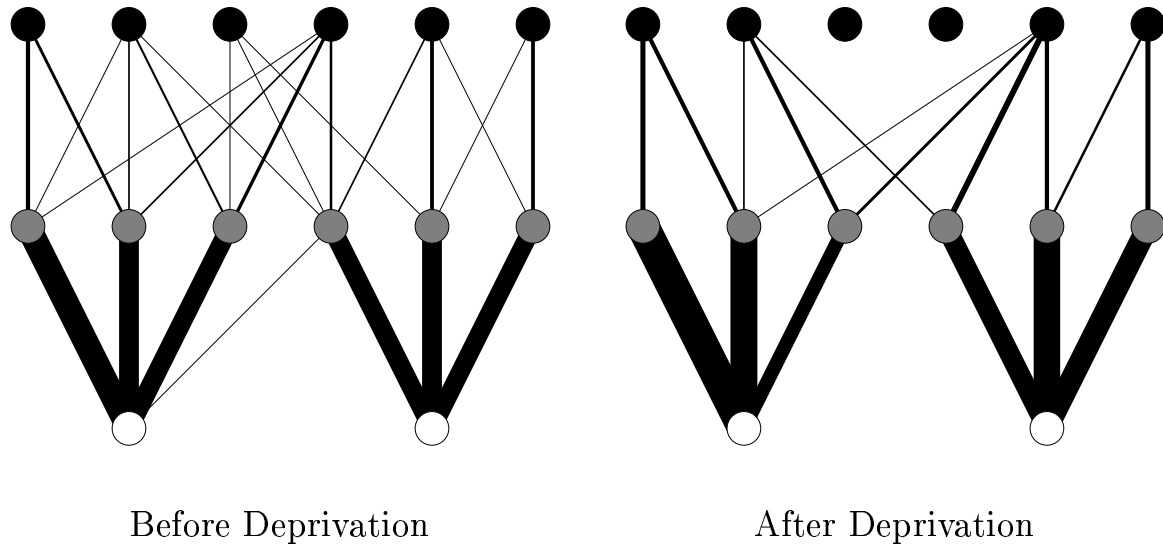


Figure 11.

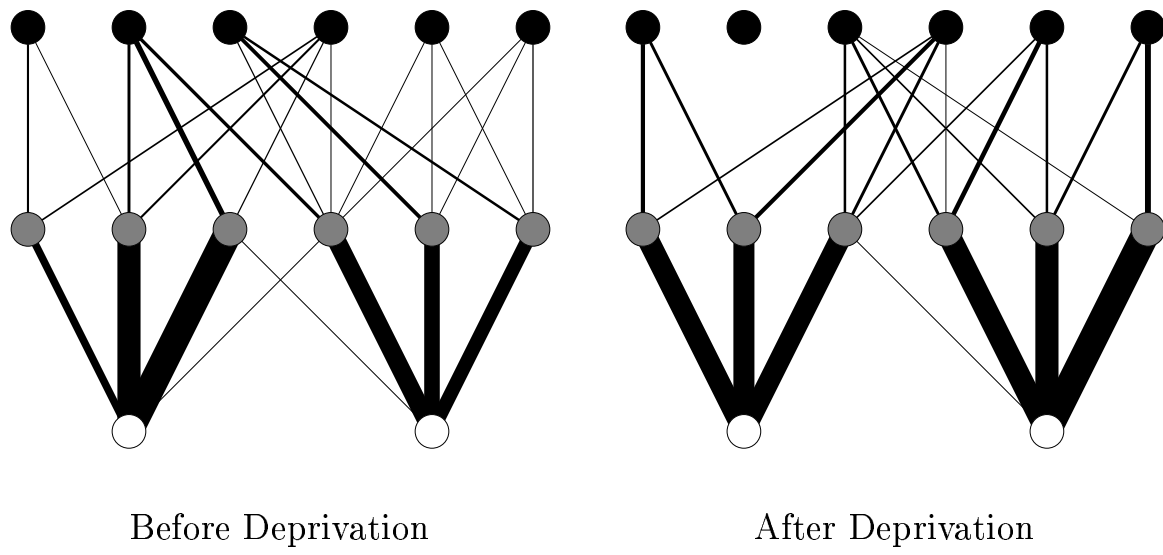


Figure 12.

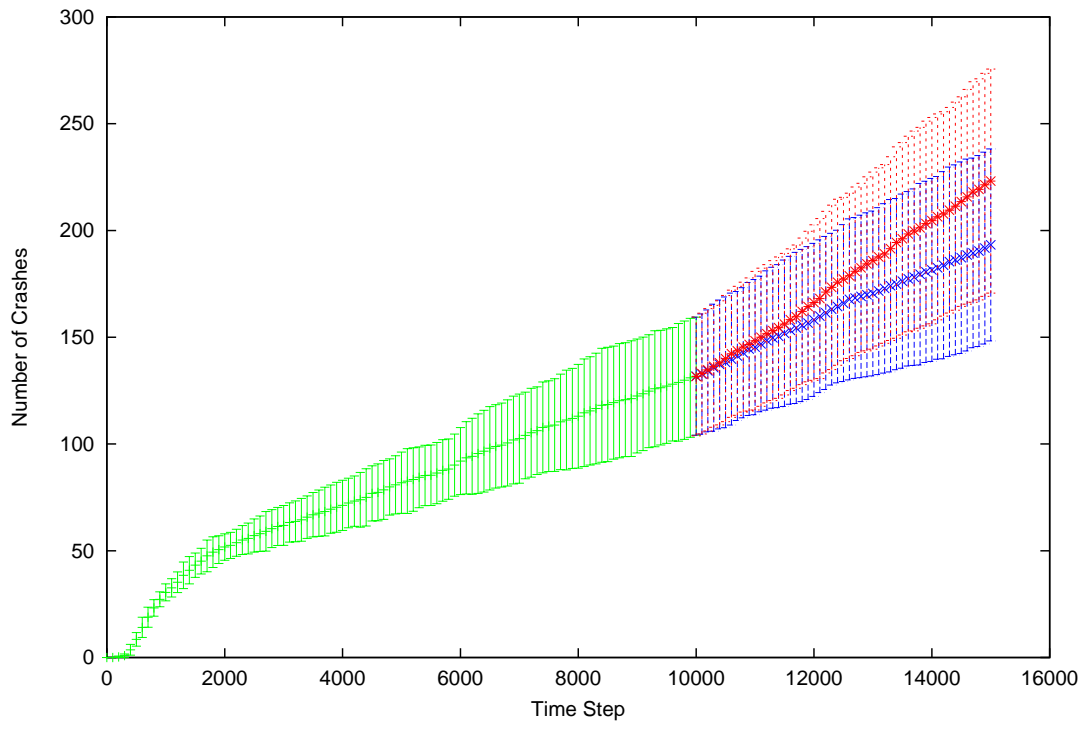


Figure 13.

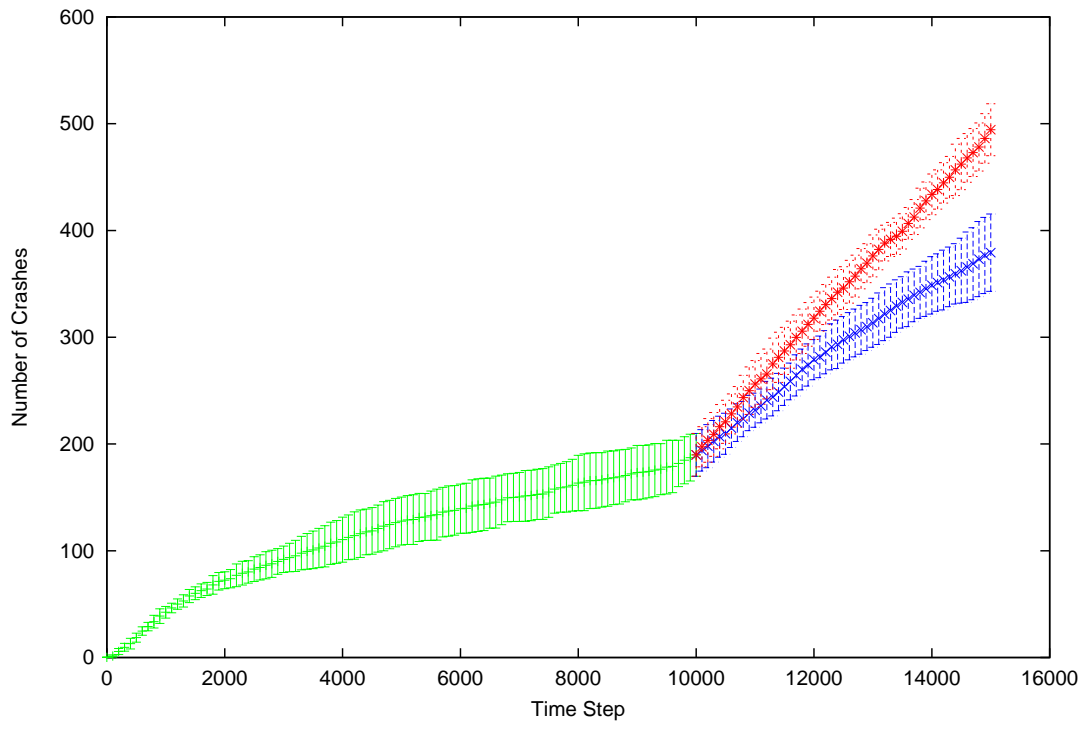


Figure 14.

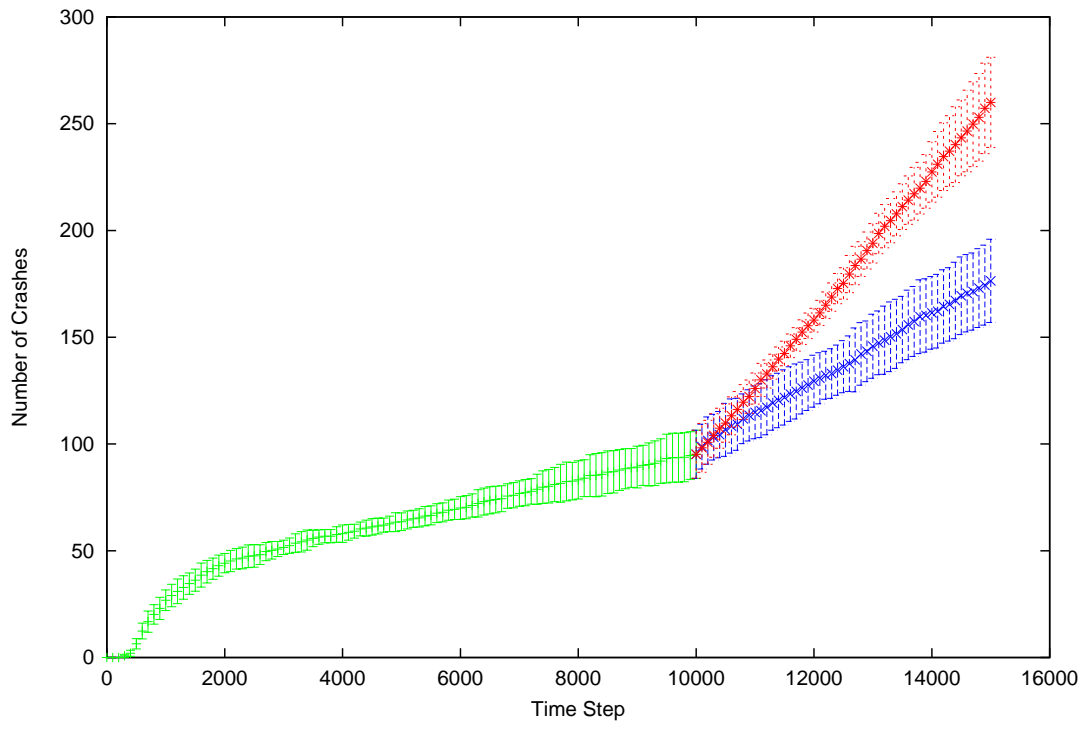


Figure 15.