Application of Systems Identification to the Implementation of Motion Camouflage in Mobile Robots

Iñaki Rañó Roberto Iglesias

Abstract

Motion camouflage is an animal stealth behaviour in which a shadower – or predator – moves in the vicinity of a shadowee – or prey – in such a way that the later perceives no apparent motion apart from the self motion. Despite some light has been shed on the control mechanism generating this pursuit strategy, it is not fully understood. Motion camouflage represents an interesting challenge in biological motion, and although simulated controllers can be found in the literature, no implementation on real robots has been done so far. This paper presents the first implementation of motion camouflage in real wheeled robots through a polynomial NARMAX model controller. The trajectories to adjust the model are generated using a heuristic approach. The NARMAX model outperforms the heuristic approach in terms of computational time and generates good camouflage trajectories in real robots and simulation. The transparency of polynomial models can also shed some light over this complex animal behaviour.

1 Introduction

Motion camouflage is a stealth behaviour observed in hover-flies (Srinivasan, 1995) and also found in territorial disputes of dragon-flies (Mizutani, Chahl, & Srinivasan, 2003). In a general pursuit situation, a predator needs to have some advantage – higher velocity or acceleration – to capture a prey (Alexander, 2003), since the prey (or shadowee in the motion camouflage scenario) will escape as soon as it detects the threat. A moving shadowee which does not perceive any threat will follow a normal path, most likely a straight line. Therefore, if the predator (or shadower) induces no optical flow on its victim's visual system while approaching, there is no need to rely on speed or acceleration superiority. Motion camouflage represents a very effective kind of pursuit behaviour exploiting the fact that depth cannot be estimated from bearing under some conditions. It is more effective since the predator chances to catch the prey are bigger, and it is more efficient because it will not spend so much energy during the pursuing. Except for the looming – the increase in size of the predator in the prey's visual system – no apparent motion is perceived.

Figure 1 shows the basic mechanism of motion camouflage. The dashed line represents the trajectory of the shadowee, and the dark dots are its position at time t_1, t_2, t_3, t_4, t_5 and t_6. If at the same time, the shadower, represented as clear circles, follows the corresponding trajectory, its motion will be camoufled. Moreover, from the point of view of the shadowee, the shadower appears as a stationary object at the focal point, the point where all the lines joining the two instantaneous positions intersect. These lines are called Camouflage Constraint Lines (CCL) as they indicate that the position of the shadower will seemingly be at the focal point from the shadowee's viewpoint. The focal point can be any point in the environment, the starting position of the shadower or a point at infinity, which produces the so called camouflage at infinity with the CCLs being parallel to each other. The trajectory followed by the shadower can approach – as presented in Figure 1 – or retreat from the shadowee, generating motion camouflage attack or retreat behaviours. It has been shown that humans can be fooled by such tactics (Anderson & McOwan, 2003a), as the peripheral visual system cannot estimate depth, and the motion camouflage effect seems to be responsible of some motorbike accidents in cross-roads (when a driver fails to detect the motion of a biker approaching an intersection). Besides the obvious military application of motion camouflage, it could be used as a deception mechanism by robots (Shim & Arkin, 2013), for instance, by flying robots to approach animals in the wild without being noticed and gather information about their behaviour. Because the predator has to stay aligned with the prey and the focal point, another potential application would be as part of a formation control strategy for multi-robot systems if three or more robots have to keep a straight segment formation (this would entail a moving focal point).

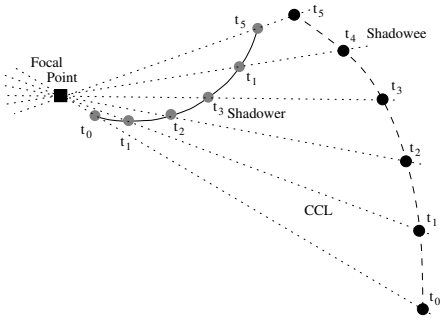


Figure 1: Motion camouflage (adapted from (Rañó, 2012))

Different works performed computer simulations of

motion camouflage, and several control mechanisms have been proposed to generate these complex trajectories. In one of the pioneer works on motion camouflage (Carey, Ford, & Chahl, 2004) an optimal control mechanism was used to generate the trajectories using a finite horizon Linear Quadratic Regulator (LQR). Assuming a constant velocity of the shadowee and no motion restrictions on the shadower, the problem was formalised as a time independent linear system, and the obtained regulator was a time dependent function. However, the assumptions of this work are not biologically plausible, since their method does not account for the heading direction and the non-holonomic restrictions to the motion of animals, which turns its implementation in real robots troublesome. A similar solution has been presented in (Rañó, 2012), where non-linear indirect optimal control is used to generate motion camouflage for a shadowee trajectory known beforehand. The optimised functional is the area of the triangle formed by the prey, the predator and the focal point. Further works (Rañó, 2013) included velocity and acceleration constraints of the shadower converting the optimal control problem into a large scale constrained optimisation problem. Despite including biologically plausible constraints to motion, these works assume perfect knowledge of the Cartesian positions of both shadower and shadowee, therefore the assumptions on the perceptual capabilities of insects are not biologically plausible.

(Anderson & McOwan, 2003b) and (Anderson, 2003) present a neural architecture to achieve simulated motion camouflage. A neural network is used to estimate the distance from the shadower to the focal point based on the position of the shadowee in the shadower's visual field. Using two other recurrent neural networks, and assuming constant linear velocity of the shadower, the motion direction and rotation is computed by the predator. Neural networks for 2D and 3D motion are trained and tested using the relative visual position of the shadowee as the only input, which corresponds to a biologically plausible setting. This work is very valuable since it shows that motion camouflage can be obtained using only the relative heading of the shadowee, but the control of the heading and motion direction are decoupled making it unsuitable as a wheeled robot controller. A curvature control law also based on biologically plausible information is derived in (Justh & Krishnaprasad, 2006; Reddy, Justh, & Krishnaprasad, 2006), where the speeds of both shadower and shadowee are assumed constant, and the heading direction is considered as a change in the curvature of the trajectories. Even though this technique provides a closed-loop expression to generate motion camouflage for any trajectory of the shadowee, its application is restricted to the case of camouflage at infinity.

Implementing motion camouflage in mobile robots is challenging as existing techniques cannot be adapted to their sensing and motion capabilities. Despite several pursuit strategies are properly understood and simple control laws can implement them (Pais & Leonard, 2010), there is no closed-form solution to the problem of generating motion camouflage trajectories with a finite focal point. This work contributes to a better understanding of this natural phenomenon by presenting the first implementation of motion camouflage on a real robot. Using polynomial NARMAX models (Chen & Billings, 1989) to learn a controller from motion camouflage training data provides a transparent way to implement it and allows interpreting the controller as opposed to, for instance, existing solutions based on neural networks. Polynomial NARMAX models have been used to implement non-linear controllers to perform different complex tasks on mobile robots like; wall following (Iglesias, Kyriacou, Nehmzow, & Billings, 2007), door traversal (Iglesias, Kyriacou, Nehmzow, & Billings, 2005) (Akanyeti, Rañó, Nehmzow, & Billings, 2008), route learning (Iglesias, Kyriacou, Nehmzow, & S.Billings, 2006) and visual object tracking (Akanyeti, Kyriacou, Nehmzow, Iglesias, & Billings, 2007). The main problem when applying this methodology to motion camouflage is to obtain data to train the controller, which, in this work, has been solved by first implementing a heuristic controller inspired by the Dynamic Window Approach to obstacle avoidance (Fox, Burgard, & Thrun, 1997). The rest of the paper is organised as follows; section 2 presents the mechanism designed to generate the training data followed by a brief review of the polynomial NARMAX approach to systems identification and the Orthogonal Parameter Estimation algorithm used to train the models. The experimental procedure to generate the train trajectories in the real robots, train and validate the polynomial controller and test it again in real robots is presented in Section 3, jointly with the experimental results. The paper ends with a discussion and further work directions in Section 4.

2 Systems Identification Applied to Motion Camouflage

Our goal is to implement a transparent motion camouflage controller in a real wheeled mobile robot to shed light on the relations between the biologically plausible perceptual inputs and the motion commands. Polynomial models are a good candidate for this, since they are simple approximators of any smooth function, but we face the problem of obtaining the training data samples. Like in programming by demonstration or imitation learning, the approach used so far to learn a NARMAX controller is to provide data generated by humans. This turns to be rather difficult in this case for two main reasons; first, the motion camouflage problem is ill-posed, and second the trajectory of the shadower – because of the non-holonomic constraints to motion – is very difficult to obtain manually. For a problem to be well-posed the solution must exist – we know motion camouflage trajectories indeed exist –, and be unique – which is not the case for motion camouflage, as for a given setup an infinite number of solutions exist. Therefore, if we would try to manu-

ally generate motion camouflage trajectories to train a learning mechanism, we should ensure that the underlying structure of all the training samples is consistent among samples, otherwise the learning mechanism will not succeed. An extreme case, for instance, would be training with trajectories for camouflage approach and retreat. On the other hand, because the controls of the shadower are the linear speed and the turning rate – not the position of the shadower –, and both signals must be properly synchronised, generating these trajectories manually is highly complex. Moreover, as we saw in the introduction, existing control mechanism to generate motion camouflage trajectories are not suitable to obtain training examples from real robots. In sum, we need a new mechanism to generate interception motion camouflage trajectories to train such a transparent model. Because the principle of motion camouflage with a finite focal point is not yet understood, nor a closed-form pursuit law exist (Pais & Leonard, 2010), any way of generating these trajectories will rely on an arbitrarily selected measure of some camouflage error E. Therefore, to obtain the training data needed for the controller we developed a heuristic error function which is evaluated and optimised like in the well know Dynamic Window Approach to obstacle avoidance (Fox et al., 1997).

Several variables have to be considered in a general motion camouflage setting; the trajectory of the shadowee, the location of the focal point, and the maximum velocities and accelerations of the shadower. The obtained controller will depend on all these parameters, which complicates the learning process. To simplify the problem we will first assume that the shadowee does not change its velocity during the process, a biologically plausible assumption if the prey does not perceive any threat. Therefore, the shadowee will follow a straight line with constant speed. We will further assume a fixed focal point located at the initial position of the shadower, and we will fix the maximum speeds and accelerations of the predator to match those of our robot hardware. Under these assumptions, the problem can be simply stated as a one step optimisation problem, i.e. finding the motion commands for the predator v^* and ω^* within the set of reachable speeds V and Ω, such that:

$$(v^*, \omega^*) = \underset{(v,\omega)\in V\times\Omega}{\arg\max}\ E(v,\omega) \quad (1)$$

where $E(v,\omega)$ is an error function designed to achieve camouflage trajectories, and all other variables and parameters fixed. The rest of the section presents the heuristic process implemented to generate the sample trajectories and the methodology used to train the polynomial model.

2.1 The Motion Camouflage Mechanism for Data Generation

To obtain the training data for the polynomial model we implemented a heuristic approach to motion camouflage inspired by the Dynamic Window Approach (Fox et al., 1997), a well known obstacle avoidance technique for unicycle type robots. The main idea behind this approach for obstacle avoidance is to define a function to optimise over the space of possible linear and angular speeds the robot can reach, that is, to maximise a function $f : (v,\omega) \to \Re$ which measures a weighted average of clearance – distance to obstacles –, target heading – progress of the robot towards the target –, and velocity of the robot. In the case of motion camouflage, a similar function can be defined to ensure the shadower moves towards the target while staying on the camouflage constraint lines.

Let us denote $\mathbf{x}_k^P = (x_k^P, y_k^P)$ and θ_k^P the position and orientation of the shadower at time k, and $\mathbf{u}_k = (v_k, \omega_k)$ the control inputs of the shadower, linear and angular speeds. The position of the shadowee at time k will be denoted as $\mathbf{x}_k^T = (x_k^T, y_k^T)$, and the fixed focal point will be denoted as $\mathbf{f} = (f_x, f_y)$. Because we assumed the shadowee follows a straight line with constant velocity ($\mathbf{v}_k^T = \mathbf{v}^T$), given an initial position $\mathbf{x}_0^T$ and the sampling time ΔT, its position at time k will simply be $\mathbf{x}_k^T = \mathbf{x}_0^T + k\Delta T\mathbf{v}^T$. It is worth noting that the proposed mechanism works for any trajectory of the shadowee provided that it is smooth enough to be predicted by the shadower. The motion of the shadower follows the unicycle model, the difference equation of its motion – the equation relating the future pose $k+1$ to the current pose and the current inputs k – can be stated as:

$$x_{k+1}^P = x_k^P + \frac{v_k^P}{\omega_k^P}\left[-\sin\theta_k^P + \sin(\theta_k^P + \omega_k^P\Delta T)\right] \quad (2)$$

$$y_{k+1}^P = y_k^P + \frac{v_k^P}{\omega_k^P}\left[\cos\theta_k^P - \cos(\theta_k^P + \omega_k^P\Delta T)\right] \quad (3)$$

$$\theta_{k+1}^P = \theta_k^P + \omega_k^P\Delta T \quad (4)$$

since the instantaneous trajectories of the shadower are arcs of circumference (Thrun, Burgard, & Fox, 2005). Our goal is to find, for every time-step k, a set of speeds (v_k^P, ω_k^P) to keep the trajectory of the shadower in the camouflage line.

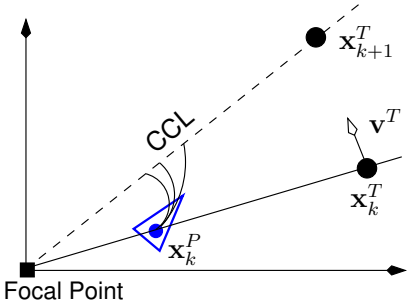


Figure 2: Shadower and Shadowee motion time step.

Figure 2 presents a motion camouflage situation, where the shadower is in perfect camouflage a time step k. Since the shadowee has a constant speed $\mathbf{v}^T$, at time

step $k+1$ its position will be $\mathbf{x}_{k+1}^T$, and the corresponding camouflage constraint line would have changed. To stay under camouflage conditions, the shadower must move following a circular arc to be at time $k+1$ in a position such that $\mathbf{x}_{k+1}^P = \mathbf{f} + \lambda(\mathbf{x}_{k+1}^T - \mathbf{f})$ for some possible values of λ, which reflects the fact that the problem is ill-posed. Given the focal point and the shadowee state – position and velocity – at time k, we can predict the position at time $k+1$ and, therefore, compute the camouflage constraint line (CCL), where the shadower should be at time $k+1$. Under our assumptions the equation of the CCL will be given by:

$$\mathbf{x} = \mathbf{f} + \lambda(\mathbf{x}_K^T + \mathbf{v}^T \Delta T - \mathbf{f}) \quad (5)$$

and, therefore, for a pair of motor commands of the shadower (v, ω), we can estimate some measure of camouflage error using equation (5) and the position prediction obtained from equations (2), (3), and (4). The estimate of the camouflage error can be then minimised to find the optimal speed commands within the reachable motion limits of the shadower.

There is a wide variety of global and local methods to optimise an error function, which in our case is defined on the continuous space of speeds. However, from a practical viewpoint, because we want to use the method in a real robot and the hardware cannot reach any arbitrary speed – reachable speeds depend on the motor controllers – it is enough to approximate the camouflage error function at a set of points in $\Re^2$, i.e. as a piece-wise constant function, which has to be fine enough to evaluate all possible robot speeds. On the other hand, a too fine sampling will increase unnecessarily the computational cost, since the robot might not be able to reach some of the speeds. Moreover, since the reachable velocities are bounded, the search space is relatively small, and a systematic search of the optimal function value can be performed like in the Dynamic Window Approach. We will further assume that the shadower moves with low speeds and, therefore any speed within the search space can be reachable in the next time-step.

Let us assume the shadower linear speed must fall on the range $v \in [v_m, v_M]$, such that $0 < v_m < v_M$, which makes the robot always have a positive velocity to ensure that the shadower does not move backwards, generating a undesirable camouflage retreat. Moreover, since the starting position is the focal point, a perfect shadower camouflage can be achieved by simply not moving, hence the strict inequality condition $0 < v_m$. Similarly, for the angular speed we will assume the range is $\omega \in [-\omega_M, \omega_M]$.

We can select now two integer numbers n_v and n_ω according to the reachable speeds of the robot, to sample the velocity space at points (v_i, ω_j) where $i = 1, \cdots, n_v$ and $j = 1, \cdots, n_\omega$, such that $v_1 = v_m < v_2 \cdots < v_{n_v-1} < v_{n_v} = v_M$ and $\omega_1 = -\omega_M < \omega_2 < \cdots < \omega_{n_\omega-1} < \omega_{n_\omega} = \omega_M$. We will partition the search space homogeneously, although, as we will see, other ways of sampling could be more appropriate for the problem at hand. The heuristic camouflage function will be computed for each of these points in $[v_m, v_M] \times [-\omega_M, \omega_M]$ and therefore the complexity of this technique is $n_v \cdot n_\omega$ times the complexity of evaluating the function. Once the function is evaluated, the optimisation takes a time proportional to $n_v \cdot n_\omega$, since an exhaustive search is performed. For each pair of speeds (v_i, ω_j), and given the current pose of the shadower $\mathbf{x}_k^P$ at time k, its next position $\mathbf{x}_{k+1}^P$ can be computed from the kinematic model, equations (2), (3) and (4). If the position and velocity of the shadowee are known (or can be estimated) the equation of the camouflage constraint line for the time step '$k+1$' can be computed, and through it, for instance, the misalignment of the shadower as a function to be minimised. We found through simulations that the Cartesian distance from $\mathbf{x}_{k+1}^P$ to the CCL is not appropriate to measure the camouflage quality of the shadower. We also found experimentally that trying to optimise only the camouflage performance is not enough for the shadower to catch the shadowee as, for instance it can follow a trajectory parallel to the motion direction of the shadowee. Moreover, the motion in the direction towards the prey can be too slow as for effectively reach it in a reasonable time. Therefore, we introduced three different terms in the function to optimise measuring: the quality of the camouflage, the distance to the prey, and the relative heading of the prey as viewed by the shadower.

The global heuristic function $E(v, \omega)$ is a linear combination of three terms $E_i(v, \omega)$ for $i = 1, 2, 3$ defined over the reachable velocity space, and normalised according to the absolute values of the function in the search domain at each time step to fall in the range $[0, 1]$. If for a given time-step the approximated function E_i is such that $E_m \leq E_i(v, \omega) \leq E_M$, all the values are scaled accordingly. These values can be easily found through an exhaustive search over the discrete space approximation, and this normalisation helps scaling the individual term contributions through the weights only. Moreover, we experimentally found that scaling the functions produces better camouflage trajectories and easier tuneable error functions. The final function to minimise is $E(v, \omega) = w_1 \tilde{E}_1(v, \omega) + w_2 \tilde{E}_2(v, \omega) + w_3 \tilde{E}_3(v, \omega)$ where $\tilde{E}_i$ represents the normalised error function and w_i corresponds to their weights, which we select to fulfil the condition $w_1 + w_2 + w_3 = 1$ to keep the global heuristic function in the range $[0, 1]$. The feasible velocity pair (v^*, ω^*) minimising this function is used as the set of commands to send to the robot, and the process repeats until the shadowee is caught, therefore no constraint is imposed to the interception time, a problem some existing techniques suffer from.

2.1.1 Motion camouflage heuristic

Different measures exist to assess the camouflage of the predicted position of the shadower, the Euclidean distance from $\mathbf{x}_{k+1}^P$ to the camouflage constraint line being one of the most obvious. This distance has a big drawback since a distance value close to the focal point

affects – from the viewpoint of the shadowee – much less the camouflage quality than that same distance close to the target. We experimentally found that the distance is not a good error function to minimise, and other works (Rañó, 2012) use, for instance, the area inside the triangle formed by the shadower, the shadowee and the focal point as the function to optimise. A more natural option would be to minimise the angular difference between the shadower and the focal point as view from the shadowee. This is in fact the error perceived by the prey when the predator approaches, since their visual system can only perceive angular distances – distance can be only estimated through motion or multiple views. To illustrate this error measure, figure 3 shows a situation in which the shadower does not exactly lay on the camouflage line. Given the focal point $\mathbf{f}$ and the positions of the shadower $\mathbf{x}^P_{k+1}$ and shadowee $\mathbf{x}^T_{k+1}$ the relative angle ϕ_{k+1} should be zero for a perfectly camouflage position. The cosinus of the angle can also be used to assess the camouflage, and can be computed from the dot product of normalised vectors $\mathbf{f} - \mathbf{x}^T_{k+1}$ and $\mathbf{x}^P_{k+1} - \mathbf{x}^T_{k+1}$. The underlying assumption is that the shadower always stays on the same side of the half plane defined by the trajectory of the shadowee, which is a requirement of motion camouflage. Therefore, the first term in the error function to optimise will be:

$$E_1(v,\omega) \quad = \quad 1 - \frac{(\mathbf{f} - \mathbf{x}^T_{k+1}) \cdot (\mathbf{x}^P_{k+1} - \mathbf{x}^T_{k+1})}{|\mathbf{f} - \mathbf{x}^T_{k+1}||\mathbf{x}^P_{k+1} - \mathbf{x}^T_{k+1}|} \quad (6)$$

where $|\cdot|$ represents the norm of the vector. According to equation (6) the error becomes zero for $\phi = 0$ and one for $\phi = \pi/2$. It is worth reminding that we experimentally found that the results obtained using this measure outperform the Euclidean distance from the shadower to the camouflage constraint line.

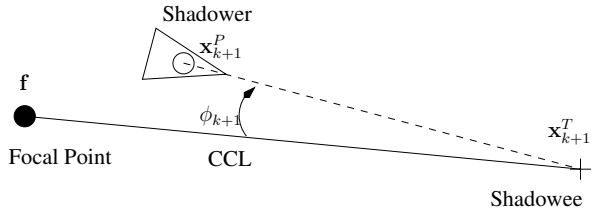


Figure 3: Camouflage error function

2.1.2 Approaching the target

As we already stated, there are two kinds of motion camouflage trajectories; approach and retreat. We are interested in implementing approaching trajectories, but this is not considered by the first error measure. In fact, we found that, under certain circumstances, the shadower moves away from the target if an additional term is not included, as further distances usually generate a smaller angular error. Therefore, the second term $E_2(v,\omega)$ in the global error can be defined based on the distance to the target along the predicted camouflage constraint line to make the shadower approach the shadowee. We simply used the normalised Euclidean distance relative to the initial distance – distance at time zero $d_0 = |\mathbf{x}^T_0 - \mathbf{x}^P_0|$– between the shadower and shadowee. Therefore the expression for the second term on the error function is:

$$E_2(v,\omega) \quad = \quad \frac{|\mathbf{x}^T_{k+1} - \mathbf{x}^P_{k+1}|}{|\mathbf{x}^T_0 - \mathbf{x}^P_0|} \quad (7)$$

which takes it maximum value ($E(v,\omega) = 1$) for $k = 0$ and decreases to zero when the shadower catches the shadowee. Ideally, in a pursuit situation, this error function will be monotonically decreasing over time, yet this cannot be ensure for motion camouflage trajectories as the distance could increase to keep a proper camouflage.

2.1.3 Heading towards the target

Although the previous contributions to the error function measuring the distance between the shadower and the shadowee tries to ensure both trajectories intercept, if the heading direction of the shadower is nearly parallel to the trajectory of the prey, the distance will decrease very slowly. This is due to the non-holonomic motion restrictions, which, in the mentioned situation, means instantaneous motion contribution perpendicular to both trajectories will be very small. This is illustrated in figure 4, which shows the simulation of an interception where the predator tries to optimise the two errors defined so far. If a heading term is included the interception occurs much earlier as the largest component of the shadower motion is towards the shadowee (cf. fig 4 and 7).

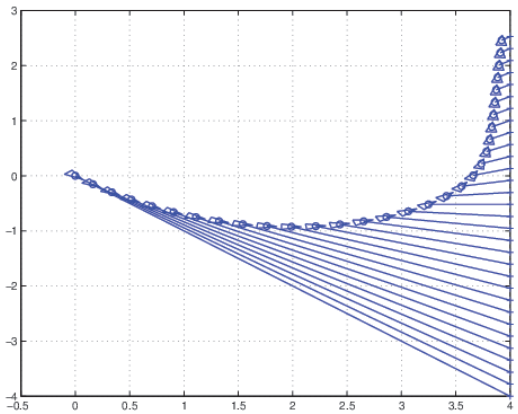


Figure 4: Heading error pointing to the target

To solve this issue we need to include an extra error factor measuring the alignment of the shadower with the predicted position of the target. Therefore, the heuristic function $E_3(v,\omega)$ measures the deviation – for each pair of potential commands – between the predicted heading of the shadower θ^P_{k+1}

and the line formed by the predicted target position $\mathbf{x}^T_{k+1}$ and shadower position $\mathbf{x}^P_{k+1}$. Figure 5 represents this error, which can be computed from the angle $\alpha = \arctan\left[\frac{y^T_{k+1}-y^P_{k+1}}{x^T_{k+1}-x^P_{k+1}}\right]$ and θ^P_{k+1} simply as $\Psi_{k+1} = \theta^P_{k+1} - \alpha$. Like in the case of the camouflage error function, we can assume that the shadower trajectory is located on one of the half-plane areas defined by the shadowee trajectory. To keep the error function normalised, we will use $E_3(v,\omega) = 1 - \cos\Psi$, which takes a zero value if the shadower is directly pointing towards the predicted position of the prey, and one when the shadowee is on either side of the shadower. It is worth noting that this function is independent of the camouflage quality but tries to keep the heading of the shadower aligned with the camouflage constraint line.

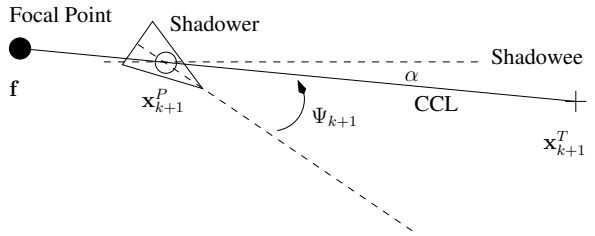


Figure 5: Heading error pointing to the target

2.2 NARMAX Models and Orthogonal Parameter Estimation Algorithm

As we mentioned, we want to identify and model the behaviour of the shadower to implement it in a real robot. We will assume that we can sample the behaviour of the shadower, i.e. we have a set of observed inputs (perceptions of the shadower) u, and outputs y (behaviour of the shadower). We aim at finding a relationship amongst past observations $[u^t, y^{t-1}]$ and the current output, y_t:

$$u^t = [u_1, u_2, ..., u_{t-1}, u_t]$$

$$y^t = [y_1, y_2, ..., y_{t-1}, y_t]$$

A successful tool for modelling these non linear dynamical systems is NARMAX (*Non-linear AutoRegressive Moving Average with eXogenous inputs*) (Chen & Billings, 1989). The general formulation of these kind of models is:

$$\begin{aligned} y_k &= F(y_{k-1}, \cdots, y_{k-N_y}, u_k, u_{k-1}, \cdots \\ &\quad \cdots, u_{k-N_u}, \epsilon_{k-1}, \epsilon_{k-2}, \cdots, \epsilon_{k-N_e}) + \\ &\quad +\epsilon_k \end{aligned} \tag{8}$$

where $F(\cdot)$ is a non-linear function, $y_{k-1}, \cdots, y_{k-N_y}$ and $u_k, \cdots u_{k-N_u}$ are the previous N_y outputs and N_u input vectors respectively. It is worth noting that we assume a multiple-input single output system, i.e. y_{k-1} is a scalar and u_{k-1} is a vector. N_u, N_y represent the number of lags in the input and output. Finally, $\epsilon_k = y_k - \widehat{y}_k$ is the *prediction error* – the difference between the training output y_k and the output computed by the model $\widehat{y}_k$, and N_ϵ is the lag of this error in the model.

A time delay in the input (d), and a dc level can easily be incorporated into the model, by rewriting Eq. 8 as:

$$\begin{aligned} y_k &= dc + F(y_{k-1}, \cdots, y_{k-N_y}, u_{k-d}, \\ &\quad u_{k-d-1}, \cdots u_{k-d-N_u}, \epsilon_{k-1}, \epsilon_{k-2}, \\ &\quad \cdots, \epsilon_{k-N_e}) + \epsilon_k \end{aligned} \tag{9}$$

(Leontaris & Billings, 1985) proved that a non-linear discrete-time time-invariant system can always be represented by the model (Eq. 8) in a region around an equilibrium point, provided that certain restrictions were fulfilled. The benefit of using this kind of NARMAX model is that it avoids the excessive number of parameters associated with Volterra series (or related Wiener series), both used to represent a large class of non-linear systems (Rugh, 1981).

In some cases, the non-linear form F(.) is known and the task of specifying the input-output relationship of the system is reduced to determining some unknown parameters. Nevertheless, for most real sampled non-linear systems their exact NARMAX models are very difficult to determine. This can be considered as *black box modelling* (Sjoberg et al., 1995), since no physical insight is available, and the chosen model will be a type of function with enough expressive power to model the system's dynamics. In the case of the NARMAX models, there are many practical examples where industrial systems have been adequately modelled by polynomial functions (Chen & Billings, 1989). Thus, we can expand Eq. 9 by defining the function F(.) to be a polynomial of degree l:

$$y(t) = \sum_{m=0}^{M} \theta_m p_m(t) + \epsilon(t) \tag{10}$$

where

$$\begin{aligned} p_0(t) &= 1 \\ p_m(t) &= y_{t-n_{y1}}, \cdots y_{t-n_{yk}} u_{t-d-n_{u1}} \cdots \\ &\quad \cdots u_{t-d-n_{uj}} \epsilon_{t-n_{\epsilon 1}} \cdots \epsilon_{t-n_{\epsilon q}} \end{aligned}$$

$$\begin{aligned} 1 \le n_{y1} \le N_y, \cdots, 1 \le n_{yk} \le N_y \\ 0 \le n_{u1} \le N_u, \cdots, 0 \le n_{uj} \le N_u \\ 1 \le n_{\epsilon 1} \le N_\epsilon, \cdots, 1 \le n_{\epsilon q} \le N_\epsilon \end{aligned}$$

The degree l of the polynomial is the highest sum of powers in any of its terms. The parameter vector θ, i.e, the coefficients of the polynomial terms can be estimated independently using the orthogonal algorithm described in (Korenberg, Billings, Liu, & McIlroy, 1988). This algorithm can be summarised in the following steps:

- The prediction errors are estimated to be zero and all the remaining parameters, which do not include ϵ are estimated. To compute the values of all the parameters but the prediction errors, it is necessary to compute an auxiliary model defined such that the terms in the model are orthogonal over the data set.

- Estimate the prediction errors, and after that the parameters associated with the prediction error terms in the polynomial model. This stage is repeated until convergence.

The determination of the model structure – which variables to include in the NARMAX model expansion described in equation (10) – is a key issue. Simply increasing the order of the dynamic terms (N_y, N_u, N_ϵ) and the order of the polynomial expansion (l) to achieve the desired prediction accuracy will in general result in an excessively complex model. This can lead to a model that overfits the dynamical system sampled data (low bias) but that generalises poorly (high variance). In general, a dynamical system has infinitely many different but input-output equivalent NARMAX models, but models with a minimal representation are usually preferred. There are several possible ways to determine which are the significant terms which should be included in the model, one of them is using the so-called *Error Reduction Ratios* (ERR). The ERR of a term of the model is the percentage reduction in the total mean-squared error – the difference between model predicted and the true system output – obtained as a result of including (in the model equation) the term under consideration. The bigger the ERR is, the more significant the term. The ERR will allow the determination of the significant terms in the model thus achieving a minimal polynomial representation able to capture the dynamics of the nonlinear process.

Hence, and to summarise this section, we can state that the NARMAX methodology breaks the modelling problem into the following steps:

- Any data set that we intend to model is first split in two sets (usually of equal size). The first is called the *estimation data set*, and will be used to calculate the model parameters. The remaining set is called the *validation set*, and is used to evaluate the model. Once this has been done, we have to set the structure of the NARMAX polynomial, i.e. we have to determine the inputs u, the output y, the lags N_u, N_y, N_ϵ, and the degree of the polynomial function.

- The polynomial model is determined using the Orthogonal Parameter Estimation algorithm to obtain the value of the coefficients.

- The NARMAX model is then tested using the validation data set. If there is no significant error between the model's predicted output and the actual output, non-contributing model terms will be removed using the error reduction errors.

- Removal of the non-relevant terms of the model: this is an iterative process that basically consists of three steps: a) the Error Reduction Ration is computed for each term. b) Those terms with the biggest ERR are removed from the model. c) The new reduced model is now validated to see whether its error is acceptable or not. If so, the error reduction error of all the remaining terms is re-calculated and a new removal takes place, otherwise the last terms removed are re-inserted back into the model equation, and the model is returned by the algorithm.

3 Experimental Procedure

To learn the polynomial controller that generates motion camouflage trajectories in a real robot, we need realistic and accurate training data. Although, in theory, the data could be obtained from simulations, the dynamic aspects of the robot motion and the intrinsic sensor noise will make a model learned from simulation useless in a real robot. Therefore, to gather the training and validation data the proposed heuristic mechanism was implemented in the real robots. Because the optimisation of the heuristic function is performed for the next step only, and its computational complexity is relatively low this technique can control the robot in real time to obtain the required trajectories. However, to confirm that we can generate accurate motion camouflage trajectories, we first show a set of proof-of-concept simulations under ideal, noise-free, conditions.

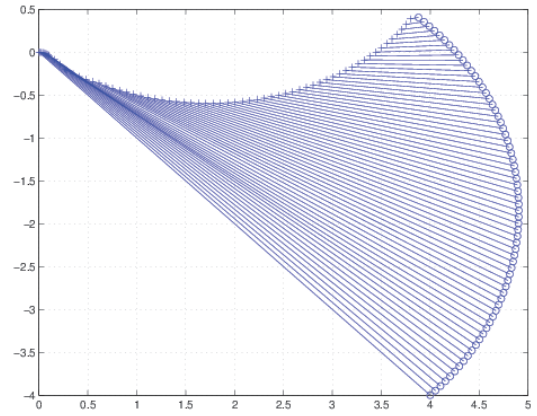
3.1 Simulated results of the motion camouflage generator

As already stated, a general motion camouflage scenario depends on several parameters (focal point position, shadowee trajectory and shadower initial pose) and because of the broad casuistry of the problem, we constrain the conditions for our case study. Although the motion camouflage generation mechanism can deal with general relative position of the focal point and the initial position of the shadower we decided to focus on the situations in which both positions coincide. In a general situation, though, the initial position of the shadower should fall on the camouflage constraint line defined by the focal point and the shadowee initial position, otherwise the trajectory would already start from a non-camoufled situation. Without loss of generality we can set the origin of a global reference system at the focal point – i.e. the initial position of the shadower.

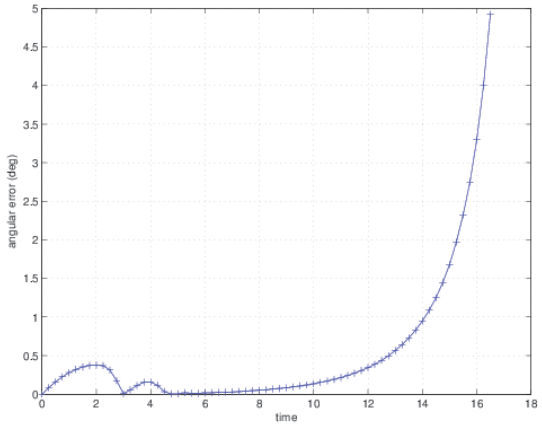
Although in the real robots we will focus on straight line trajectories of the shadowee, we ran simulations of the motion camouflage generator algorithm for different trajectories. Straight line trajectories can be modelled as a simple integrator, yet in our simulations the prey also follows the unicycle motion model:

$$\dot{x}^T = v^T \cos\theta^T \quad (11)$$
$$\dot{y}^T = v^T \sin\theta^T \quad (12)$$
$$\dot{\theta}^T = \omega^T \quad (13)$$

where (x^T, y^T, θ^T) and (v^T, ω^T) are the pose and velocity commands of the shadowee respectively. A straight line trajectory can be obtained from this model for some initial conditions $(x_0^T, y_0^T, \theta_0^T)$ simply by setting $\omega^T = 0$ and fixing v^T.



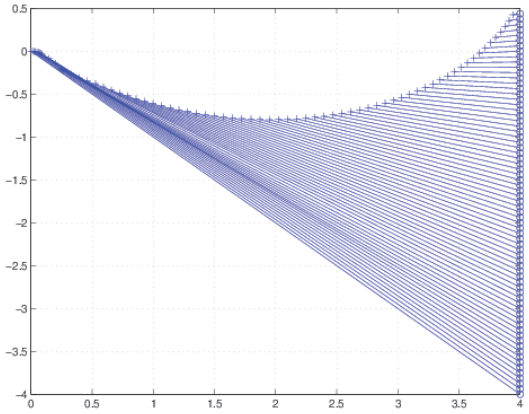
(a) Motion camouflage trajectory



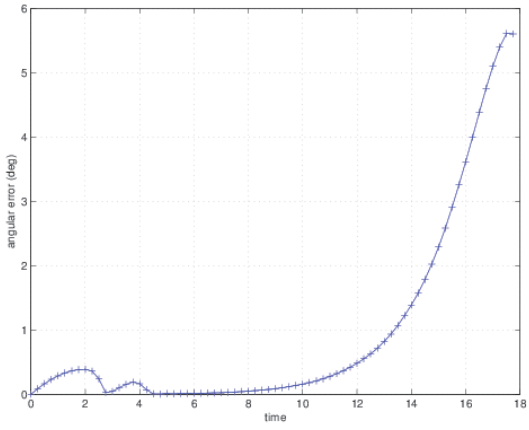
(b) Trajectory angular error

Figure 6: Simulated motion camouflage trajectory for the shadowee following a circular arc

Figure 6 shows the results of the simulation of a motion camouflage scenario with the shadowee following a circular arc trajectory. Both, the linear and angular velocities were fixed to $v^T = 0.3$ m/s and $\omega^T = 0.1$ rad/sec, with an initial pose of $(x^T, y^T, \theta^T) = (4, -4, \pi/4)$ and a sampling time interval of $\Delta T = 0.25$ s. Figure 6(a) shows the trajectories of the shadower and the shadowee, with the instantaneous positions joined by lines, which, ideally should all contain the origin of the Cartesian coordinate system (the focal point). The shadower is represented as the '+' sign in the figure, and the shadowee by '$\circ$'. To illustrate the quality of the camouflage trajectory, figure 6(b) shows the absolute angular error as a function of time. As the figure shows the error increases as the shadower approaches the target but it stays below 5°. Obviously, as the shadower gets close to the shadowee a small Cartesian distance to the Camouflage Constrain Line corresponds to a larger angular error, which makes the angular error increase. This is an effect of discretising the velocity space, since the sampling of the Cartesian positions forms a triangular mesh on the $x - y$ plane. As the shadower tries to minimise the distance to the shadowee, it will always opt for high speeds, which represent larger distances between samples and, therefore, larger camouflage errors. However, the selected sampling produces acceptable angular deviations.



(a) Motion camouflage trajectory



(b) Trajectory angular error

Figure 7: Simulated motion camouflage trajectory for the shadowee following a straight line

Figure 7 shows the motion camouflage simulation when the shadowee follows a straight line trajectory with a constant forward speed of $v^T = 0.25$ m/s. The plot of the error evolution over time (figure 7(b)) shows that the error is bounded by 6° and it also increases when the shadower approaches the shadowee due to

the velocity space discretisation. In both simulations a threshold distance of 5 cm is defined as the stop criterium – i.e. when the distance between shadower and shadowee is smaller than this threshold – which is a very small distance compared to the initial distance between the two agents. The initial pose of the shadower is $(x^P, y^P, \theta^P) = (0, 0, 0)$ for the two simulations, which coincides with the focal point. As we can see in the horizontal axis of the error plots, the interception time is different for each simulation and depends on the specific trajectory followed by the shadower.

3.2 Data collection on the real robots

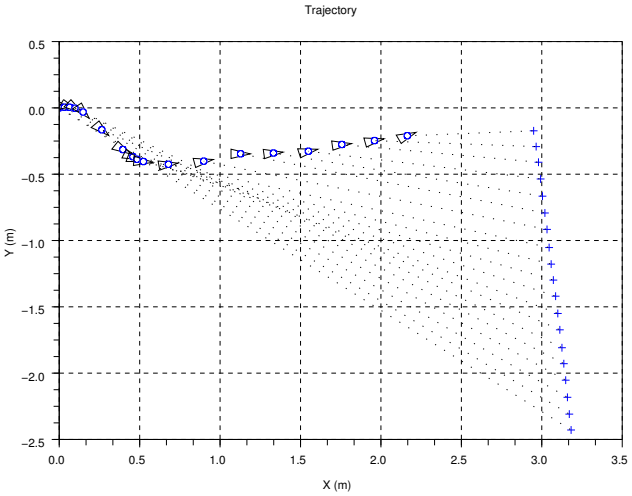
Once the heuristic mechanism to obtain motion camouflage was tested in simulation, we implemented it in a dual-drive Pioneer 3 robot. Our robots are equipped with different sensors; a digital camera mounted on a pan & tilt unit, a ring of sonar transducers, two microphones, odometry sensors, and a SICK laser scanner. The odometry and the laser sensor could be used by the shadower to localise itself and the shadowee in the robot arena, while the shadowee velocity could be estimated from the laser readings, for instance, through a Kalman filter. However, odometry is unreliable since errors accumulate as the robot moves (Borenstein & Feng, 1996) and we need accurate data to train our controller. An option to correct these errors would be to localise the robot through landmarks also detected with the laser scanner, but in the motion camouflage process the region in which both agents move should be free of obstacles. This makes difficult the problem of placing landmarks to localise the robot, since they must be far enough not to interfere with the trajectories, therefore reducing the localisation accuracy.

Our lab facilities include a VICON tracking system that provides fast and accurate 3D information about the Cartesian positions and orientations of objects within the robot arena. It obtains readings at a frequency of 60 Hz and with a position accuracy of passive markers in the order of the millimetre. We used the tracking system to obtain the position and orientation of the robots, but because of their relative low speeds the information was subsampled using a sampling period of 0.3 s. The uncertainty the VICON system provides is negligible for the problem at hand and it is bounded along the whole trajectory, i.e. it does not increase with time. This external visual tracking system clearly outperforms any robot sensor based existing technique to estimate the position and orientation of the robots, both in terms of accuracy and sampling time. Passive infra-red reflecting markers were placed on the Pioneer 3 robots to be uniquely identified by the VICON tracking system, and the centre position and orientation of both robots were obtained. From the position of the shadowee its velocity – we assume constant – can be easily estimated. This provided accurate position, velocity and orientation information needed to apply our motion camouflage generation mechanism and obtain reliable training data. Therefore, the heuristic method based control program to generate the motion camouflage training data was fed with the information provided by the VICON tracking system through the wireless network available in our lab, while the expected position of the target was obtained using the estimated velocity. The program stored in a file all the sensor information (sonars, laser and odometry) and the issued motor commands, to later train the NARMAX controller.

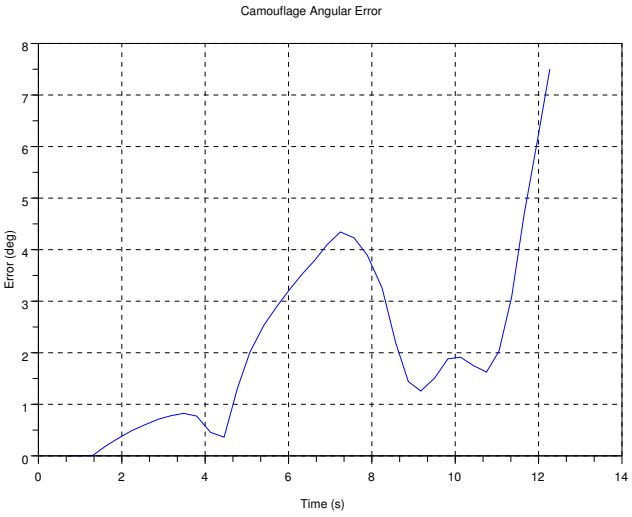
To run the experiments we placed both robots in the centre of the area covered by the VICON tracking system to maximise their visibility and ensure the trajectories could be tracked accurately. The initial relative headings of the robots was 90° and the origin of the reference system of the VICON was moved to coincide with the focal point (also the initial pose of the shadower). The x-axis of the reference system was chosen to be the initial heading of the shadower and the initial position of the shadowee in the reference system was approximately $x^T \approx 3$ m and $y^T \approx -2.5$ m with its heading roughly corresponding with the 'y' axis (90°). This configuration represents an initial distance between the shadower and the shadowee of around 4 m, within the detection range of the laser scanner. The speed of the shadowee for all the tests was fixed to 0.2 m/s, but instead of being fed directly in the algorithm, as already stated, it was estimated from the tracking information at every time-step to account for its potential variation. The maximum speed of the shadower was set to 0.35 m/s, and, because no safety mechanism to avoid collisions between the two robots was implemented, the program was manually stopped right before the interception occurred for each trial.

Figure 8(a) shows one of the collected training camouflage trajectories, where the shadower is drawn as small triangles – representing the heading – with a circle inside – representing the position – and the shadowee simply as '+' signs, since its heading is not relevant. The 'x' axis is the initial orientation of the shadower, and the dotted lines join the position of both robots obtained by the VICON tracking system, while for clarity purposes only one out of two positions of the whole trajectory is shown in the figure. Figure 8(b) shows the corresponding camouflage error for the trajectory, which has a maximum value smaller than 8°. Although the error seems similar to the one obtained in the simulated trajectories (cf. figures 6(b) and 7(b)), it is worth noting that, due to the size of the robots, the distance between them does not get as close as in the simulations, and therefore the angular error is larger along the trajectory. The positions shown in the figure correspond to the centre points of the robots, which, because of their physical size, is about 0.5 m at the interception point, ten times larger than the 5 cm threshold selected for the simulations.

Using these initial conditions, and the constant shadowee speed, we collected data for twenty-five sample camouflage trajectories with a sampling time of 0.3 s. These trajectories were used to obtain (train and validate) the NARMAX polynomial models. The stored position information of both robots jointly with the laser readings of the shadower helped identifying the



(a) Motion camouflage trajectory



(b) Trajectory angular error

Figure 8: Sample training trajectory

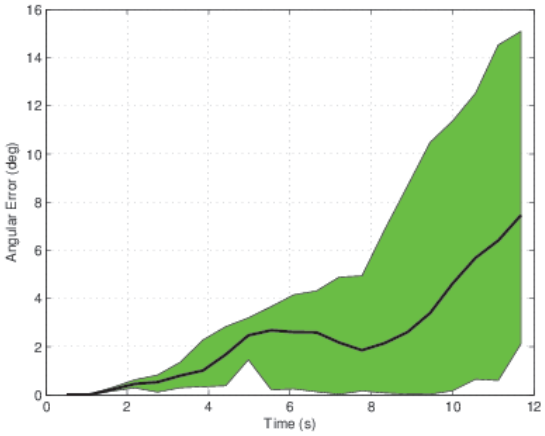


Figure 9: Overall errors on the training and validation data

relative position of the shadowee. The average initial distance among the training dataset – obtained from the tracking system – was 4.14 m with a standard deviation of 9 cm, while the final distance between the two robots was 0.63 m on average with a standard deviation of 4 cm. The average interception time for all the train samples was 13.28 seconds. Figure 9 shows the camouflage errors for the training and validation trajectories as a function of time, i.e. the trajectories obtained from the robot experiments using our heuristic approach. The plot shows the average angular error as a function of time enclosed between the maximum and minimum errors for all the trajectories, and the error plot is computed for all the trajectories only up to the lower interception time.

3.3 Model Generation and Testing

Our goal is to obtain a motion camouflage controller, through system identification, that performs well with plausible biological inputs. However, the heuristic mechanism for data generation relies on Cartesian coordinates and velocities to generate the camouflage control variables. Clearly, insects do not use absolute Cartesian coordinates to drive its camouflage behaviour. Since the aim of this work is to generate a biologically plausible controller, we defined new inputs for the NARMAX model different from the those used by the heuristic controller. Therefore, the inputs used to train the controller were computed from the robot's on-board sensors extracted from the data stored for the heuristic method runs. To obtain good training data, we performed experiments under conditions ensuring the generated model will capture the essence of the behaviour with sufficiently realistic noise, and under our assumptions (see section 2). For instance, the laser scanner obtains relative angular positioning and distance, which simplifies the whole sensorial information processing, whilst providing a realistic noisy signal. The relative angular position and distance to the shadowee as seen by the shadower are biologically plausible inputs for the controller. Since the shadower moves in a obstacle free area it is easy to detect the pattern of the shadowee. A blob search algorithm implemented to find the position of the shadowee in the laser readings provided the information necessary to feed the controller. Therefore, a simple pattern search on the difference between consecutive laser readings suffices to detect the shadowee.

Figure 10 represents the selected input variables for the controller. The first sensorial reading of the shadower robot when the motion camouflage process starts was used to compute initial values for d_0 and ϕ_0, the initial distance to the target and relative angle (which for the first step coincides with the absolute angle). As the camouflage trajectory evolves, the controller input variables will be: i) the quotient between the current and the initial distances $\frac{d_k}{d_0}$ and ii) the difference between the view angles of the shadowee $\Delta\phi_k = \phi_k - \phi_0$. These variables are both relative to the shadower pose, therefore, they can be easily computed from its sensors. From the blob, obtained using the above mentioned

blob detector at each time step, the distance and relative angles are derived and used to compute the input for the controller.

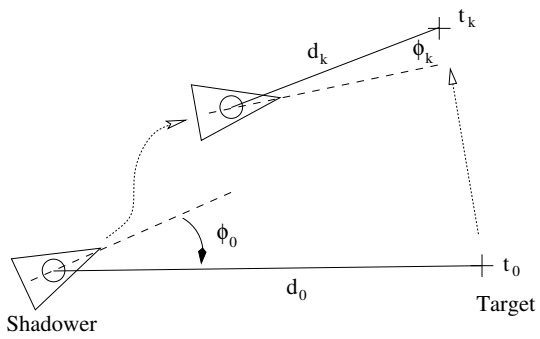


Figure 10: Input variables used by the controller

Since our heuristic controller computes the linear and angular velocities, two NARMAX controllers must be obtained, one for each output variable. They were trained and validated using the data from the camouflage trajectories presented in section 3.2 after computing the corresponding input variables. Our implementation of the Orthogonal Parameter Estimation algorithm splits the training dataset into two equal parts one for learning and one for validation of the model. However, contrary to the methodology presented in section 2.2, the testing phase will be carried in the real robot as we will see later. The polynomial controllers obtained are:

$$v_k = 0.393 - 0.190\Delta\phi_k - 0.007\frac{d_k}{d_0} - 0.108\Delta\phi_k^2 - 0.274\left[\frac{d_k}{d_0}\right]^2 + 0.331\Delta\phi_k\frac{d_k}{d_0} \quad (14)$$

$$\omega_k = 0.486 - 0.148\Delta\phi_k - 2.825\frac{d_k}{d_0} + 4.941\left[\frac{d_k}{d_0}\right]^2 - 2.837\left[\frac{d_k}{d_0}\right]^3 + 2.808\frac{d_k}{d_0}\Delta\phi_k - 2.877\Delta\phi_k\left[\frac{d_k}{d_0}\right]^2 + 0.117\frac{d_k}{d_0}\Delta\phi_k^2 \quad (15)$$

The algorithm used to obtain the NARMAX model selected only the relevant terms to generate the behaviour according to the training data trajectories. For instance, the model corresponding to the angular velocity had initially 12 terms which were reduced to 8 after the removal of the non-relevant monomials using the Error Reduction Ratios. Figures 11 and 12 show the outputs of the validation dataset – angular and linear velocities respectively –for all the validation trajectories (chained one after the other) superposed with the prediction of the NARMAX model. Therefore, the x axis represents the index of the validation data point – for all the dataset – and the y axis the angular and linear speeds. These models were obtained with parameters $N_u = 0$, $N_y = 0$, $N_e = 0$, and the initial

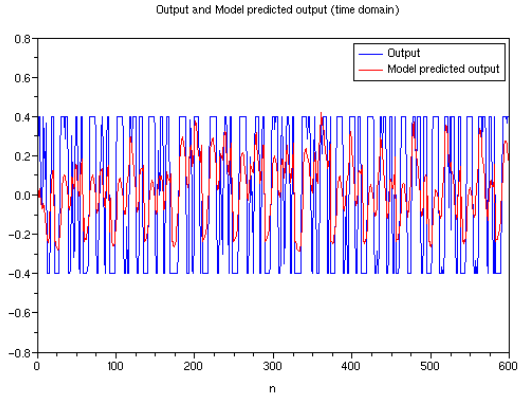


Figure 11: Outputs of the validation data and prediction of the NARMAX model for the angular velocity

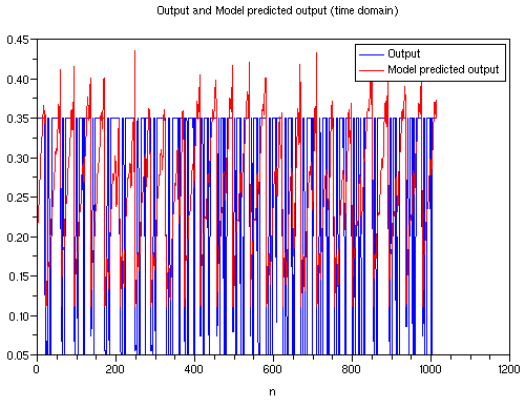


Figure 12: Outputs of the validation data and prediction of the NARMAX model for the linear velocity

degree of the polynomial was 3. Units of all variables are SI (International System), therefore, angles are in radians and distances in metres. Likewise the turning rate is obtained in rad/s and the forward speed in m/s.

Obviously the computation time needed to generate a pair of velocities using the polynomial is much less than using the heuristic algorithm, since the late has to perform a function evaluation and a search over the velocity space. In fact the velocities are computed from a fixed number of additions and products, much lower than the exhaustive evaluation and search in the $n_v \times n_\omega$ discretised space for the heuristic method.

Once the pursuer-relative control models of equations (14) and (15) were obtained, we used them to control the shadower, while the target robot performed straight-line motion. We conducted 37 test runs with the Pioneer 3 robots and a similar experimental setup as for the data collection, i.e. both robots started around 4 m apart from each other with initial orthogonal heading directions. Specifically, the initial distance between the two robots was 4.05 m on average, with a standard deviation of 6 cm, and the final distance was 0.67 m. The average interception time of the test trajectories was 11 s with a standard deviation of 0.45 s.

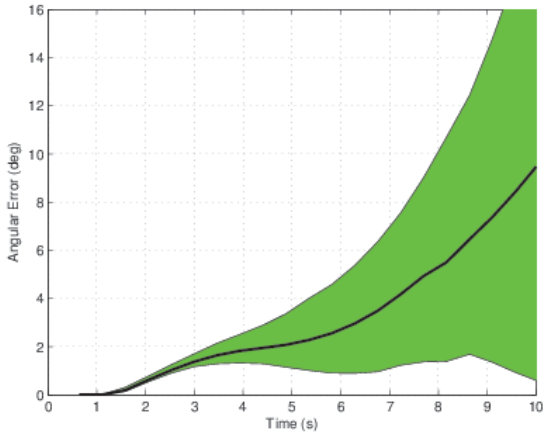
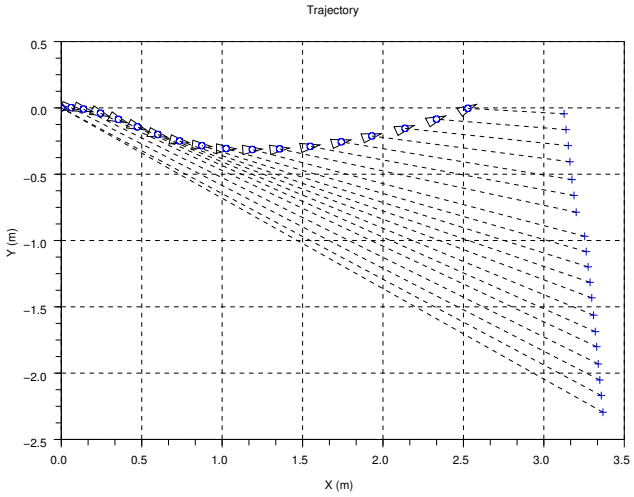
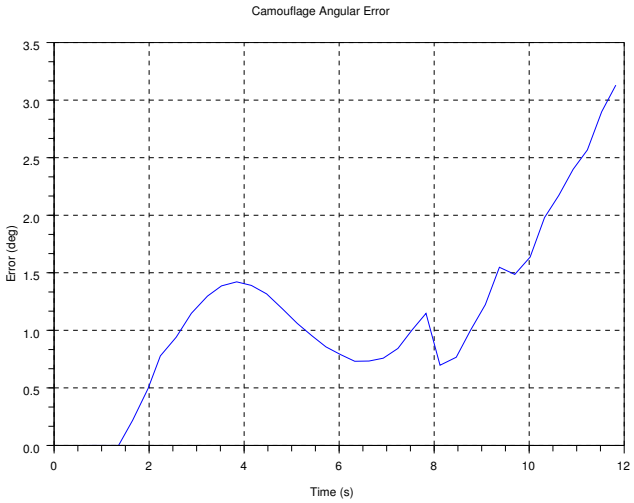


Figure 13: Overall errors on the experiments with the learnt NARMAX controller



(a) Motion camouflage trajectory



(b) Angular trajectory error

Figure 14: NARMAX controller test trajectory

Figure 13 shows the average error for all the test trajectories with the maximum and minimum error bounds. As it can be seen, the errors of the NARMAX controllers are larger than the heuristic method. This is normal since the polynomial controller is a truncated non-linear approximate of the solution. The focal point was also the initial position of the shadower, and the shadowee velocity was set to 0.2 m/s. The VICON system was used to record the exact trajectories of both robots and measure the camouflage performance using the angular error. Figure 14 shows one of the experiments using the NARMAX controller, for which the median error angle is $-1.8°$, with a confidence interval of $[-1.3°,-6.5°]$ at the 5% significance level. Figure 14(a) shows the trajectories of the shadower and shadowee as collected from the VICON system, which as it can be seen is smoother than the train trajectory depicted in Figure 8(a). The fact that the trajectories of the NARMAX controller are smoother is the consequence of the removal of some terms of the polynomial controller by the train algorithm, as higher order terms in the polynomial represent abrupt changes in the controller. The orthogonal parameter estimation algorithm was able to identify the relevant terms to be included on the controller and capture the essence of motion camouflage. Like for the train data trajectories the test runs lasted about 13 sec and the error increases over the time. This effect cannot be avoided as it is a feature of the heuristic algorithm and, therefore, of the training data.

Figure 15 represents four frames corresponding to a video recorded with a camera from the shadowee point of view. The pictures show how the shadower appears always in the corresponding camouflage line which lays approximately to the left of the centre of the door. All in all, for the 37 experiments performed using the NARMAX controller the median angular error was $-5.3°$ ($[-5.1°,-5.5°]$, $p = 0.05$).

3.4 Simulation of the NARMAX model and discussion

We further implemented the polynomial model, equations (14) and (15), in simulation to test their working limits as a motion camouflage controller. The rationale for this is that simulating the model is less time consuming than running it on the real robot. Figure 16 shows the results of simulating the model for initial conditions similar to those in the training set. As figure 16(a) shows the controller can extrapolate camouflage behaviour even for unseen situations like the shadower and shadowee being closer than 0.6 m (the limit imposed by the physical size of the robots). Despite the angular error being larger than the one obtained in simulation through the heuristic mechanism (cf. figures 16(b) and 7(b)), as we argued earlier there is a clear computational advantage on using the polynomial model.

From the model equations we see that both variables, the distance and relative angle contribute to the linear and angular speeds of the learnt motion camouflage

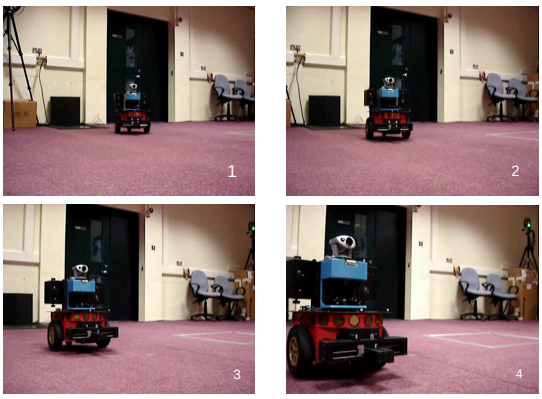
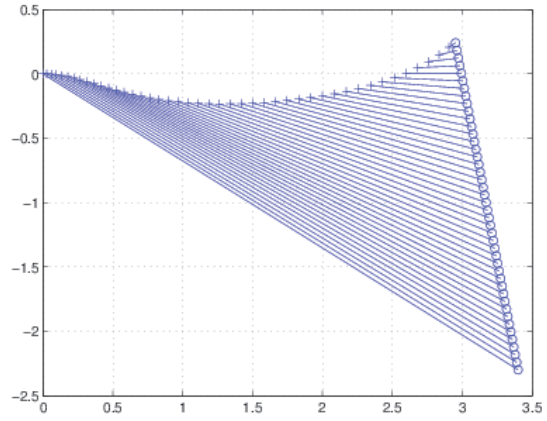
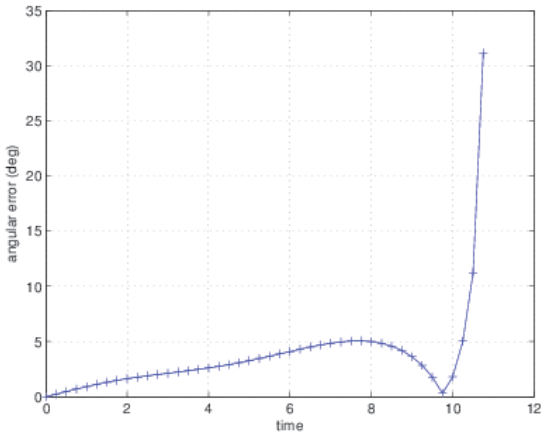


Figure 15: Images from the shadowee point of view



(a) Motion camouflage trajectory



(b) Angular trajectory error

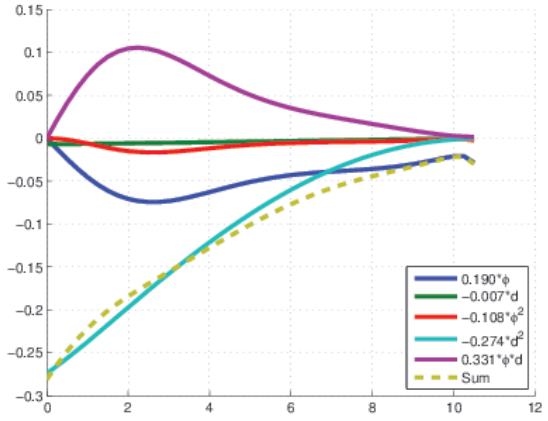
Figure 16: Simulated NARMAX controller

controller. However, as the shadower approaches the shadowee, the terms containing $\frac{d}{d_0}$ represent a smaller contribution to the speeds. Figure 17 shows, as a function of time, the contributions of the individual terms of the polynomial model – except the DC term – for the simulation presented in figure 16, and the sum of all the monomials is shown as a dashed line. From the figures we can see that the distance between the shadower and the shadowee decreases approximately as a linear function of time (see monomial '$-2.825d$' in figure 17(b)), an interesting result as both agents are moving. This also means that close to the interception point the terms involving the distance will have less importance (eventually vanishing when the predator captures the prey). Figure 17(a) shows how the forward speed (the sum of the dashed line and the DC component) of the shadowee increases with time and its profile is quite similar to the monomial on d^2. The profile of the angular velocity is slightly more complex as it increases or decreases with time in different parts of the trajectory and there is no simple way to have a good approximation with fewer terms. In any case we can see that both controllers can be generated at most with polynomials of order three, and the selected inputs allow implementing motion camouflage. This matches the hypothesis on how motion camouflage is generated in nature, as the angular direction to the shadowee can be directly obtained by the shadower, while the distance can be estimated using the visual apparent size of the prey. The Orthogonal Parameter Estimation algorithm selected a parsimonious model to generate motion camouflage, and it shows that this behaviour can be achieved with relatively simple controllers as the degrees of the polynomials are smaller or equal than three.

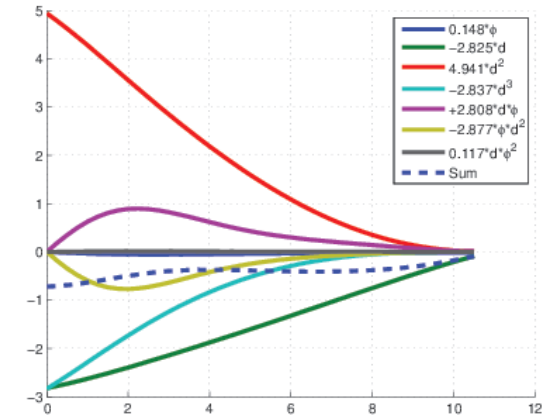
Param.	Train value	Min. range	Max. range
Heading	90°	69°	96°
Speed (m/s)	0.2	0.185	0.25
Dist. (m)	4.14	1.0	4.5

Table 1: Training parameters and working limits of the NARMAX controller

To evaluate the working limits – the generalisation – of the NARMAX model, we performed simulations changing the parameters relative to the shadowee fixed during the robot experimental runs. These parameters are the initial heading of the shadower, its forward velocity and the initial distance to the shadower. As table 1 shows the approximate values to obtain the training data were 90°, 0.2 m/s, and 4.14 m, respectively. We fixed the proximity threshold distance – final distance between the shadower and the shadowee to 0.6 m and considered that a good generalisation for the camouflage trajectory had a bounding angular error (see section 2.1.1) of 16°. Based on these values we changed the parameters incrementing and decrementing their value by a small amount until the simulated trajectory error was above 16°. Table 1 summarises the resulting working limits of the polynomial controller. It is worth noting that although the standard deviation of the initial distance between the shadower and the shadowee was 0.09 m for the training data, the controller can generalise to large range of distances (specially for shorter distances). The table also shows that generalisation ability with respect to the initial direction of



(a) Monomial contribution for the v controller



(b) Monomial contribution for the ω controller

Figure 17: Simulated NARMAX controller contributions

the shadowee is not symmetric, since it extends almost 20° towards lower headings. In these situations the shadowee approaches the shadower and the interception occurs faster but without increasing camouflage the error.

4 Conclusions and Further Work

This paper presents the first implementation of motion camouflage behaviour in real robots through a non-linear polynomial controller learnt from real data using the Orthogonal Parameter Estimation algorithm. A computationally expensive heuristic mechanism inspired by the Dynamic Window Approach to obstacle avoidance was developed to obtain the training data, while the camouflage accuracy of the learnt controller was close to the heuristic approach (with much less computational requirements). The controllers were obtained as polynomials of degree smaller or equal than three, which indicates motion camouflage with a finite focal point can be generated through relatively simple mechanisms.

Contrary to existing optimal control techniques (Rañó, 2012) (Carey, 2007) the whole trajectory does not need to be know beforehand and the inputs of the obtained controller are biologically plausible. Whilst these optimal control based methods require accurate knowledge of the whole trajectory to compute the velocity commands as a function of time, our method requires only a prediction of the position of the shadowee at the next time-step. This requirement is also eliminated in the polynomial controller since speeds are generated using current sensor inputs. Existing closed-loop mechanisms to generate motion camouflage either focus on camouflage at infinity or use artificial neural networks, which do not provide a direct inside view, for instance, on the controller complexity. The view that to generate motion camouflage trajectories the control of the linear velocity was a function of the distance to the target only, and the control of the angular velocity depends uniquely on the relative prey direction is presented in (Rañó, 2012). Our approach seems to contradict that view since both learnt controllers depend on both input variables.

A well know drawback of the Orthogonal Parameter Estimation algorithm is its sensitivity to the order in which the controller variables are presented. This means that the obtained controller is parsimonious for the selected order, but an even simpler motion camouflage controller could exist. We plan to investigate this in the future to find the minimal set of inputs and monomials that can generate this complex animal behaviour.

The performance of our heuristic implementation of motion camouflage degrades as the shadower approaches the shadowee because the discretisation on the velocity space induces a discretisation on the Cartesian positions the predator can reach. This was reflected in the final part of the error plots, where the angular error always increased. This obviously affects the polynomial NARMAX controller, which presents the same performance degradation for close distances between the two agents. To solve this problem, in the future, we plan to introduce a distance dependent discretisation to allow a finer Cartesian sampling and reduce the angular error for close distances. Even though the used input variables of the controller are biologically plausible, as a first approach, we disregarded the possibility of using landmarks on the environment to localise the shadower. As a future line of work we plan to include information of potential landmarks in the polynomial model, such that the controller will be more robust to changes in the initial conditions – position and velocity – of the shadowee. This would provide a broader view of the situated motion camouflage process.

Acknowledgments (In memoriam)

The authors would like to dedicate this work to the memory of Ulrich Nehmzow, who was involved in, and made essential contributions to the work reported in this paper. He was an enthusiastic researcher who made extraordinary contributions in the field of cognitive robotics.

References

Akanyeti, O., Kyriacou, T., Nehmzow, U., Iglesias, R., & Billings, S. (2007). Visual task identification and characterization using polynomial models. *Robotics and Autonomous Systems*, *55*, 711-719.

Akanyeti, O., Rañó, I., Nehmzow, U., & Billings, S. (2008). A proposal of a methodology for the analysis of robot-environment interaction through system identification. In *Proceedigs of taros 2008*.

Alexander, R. M. (2003). *Principles of animal locomotion*. Princeton University Press.

Anderson, A. (2003). *Sensory motor neural systems for a predatory stealth behaviour camouflaging motion*. Unpublished doctoral dissertation, Computer Science, Queen Mary University of London.

Anderson, A., & McOwan, P. (2003a). Humans deceived by predatory stealth strategy camouflaging motion. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *270*(Suppl 1), S18.

Anderson, A., & McOwan, P. W. (2003b). Model of a predatory stealth behaviour camouflaging motion. *Proceedings Royal Society B*, *270*, 489-495.

Borenstein, J., & Feng, L. (1996). Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, *12*(6).

Carey, N. E. (2007). *Biomimetic strategies for motion camouflage*. Unpublished doctoral dissertation, Australian National University.

Carey, N. E., Ford, J. J., & Chahl, J. S. (2004). Biologically inspired guidance for motion camouflage. In *Proceedings asian control conference 3* (Vol. 3, p. 1793-1799).

Chen, S., & Billings, S. A. (1989). Representations of non-linear systems: The NARMAX model. *Control*, *49*, 1013-1032.

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, *4*(1), 23-33.

Iglesias, R., Kyriacou, T., Nehmzow, U., & Billings, S. (2005). Robot programming through a combination of manual training and system identification. In *Proceedings of the 2nd european conference on mobile robots* (p. 158-163).

Iglesias, R., Kyriacou, T., Nehmzow, U., & Billings, S. (2007). Task identification and characterization in mobile robotics through non-linear modelling. *Robotics and Autonomous Systems*, *55*(4), 267-275.

Iglesias, R., Kyriacou, T., Nehmzow, U., & S.Billings. (2006). Route training in mobile robotics through system identification. In *Proceedings of the 15th international conference on computer and information science and engineering* (p. 181-186).

Justh, E. W., & Krishnaprasad, P. S. (2006). Steering laws for motion camouflage. *Proceedings Royal Society A: Mathematical, Physical and Engineering Sciences*, *462*, 3629-3643.

Korenberg, M., Billings, S., Liu, Y., & McIlroy, P. (1988). Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control*, *48*(1), 193-210.

Leontaris, I., & Billings, S. A. (1985). Input-output parametric models for non-linear systems. part i and part ii. *International Journal of Control*, *41*, 303-344.

Mizutani, A., Chahl, J., & Srinivasan, M. (2003). Insect behaviour: Motion camouflage in dragonflies. *Nature*, *423*, 604.

Pais, D., & Leonard, N. E. (2010). Pursuit and evasion: evolutionary dynamics and collective motion. In *Proceedings of aiaa guidance, navigation, and control conference on aiaa* (pp. 1–14).

Rañó, I. (2012). An optimal control strategy for two dimensional motion camouflage. *Biological Cybernetics*, *106*(4-5), 261-270.

Rañó, I. (2013). Direct collocation for two dimensional motion camouflage with non-holonomic, velocity and acceleration constraints. In *Proceedings of the ieee conference on robotics and biomimetics*.

Reddy, P. V., Justh, E. W., & Krishnaprasad, P. (2006). Motion camouflage in three dimensions. In *Ieee conference on decision and control* (p. 3327-3332).

Rugh, W. J. (Ed.). (1981). *Nonlinear systems theory: The volterra/wiener approach* (No. 0801825490). The Johns Hopkins University Press.

Shim, J., & Arkin, R. (2013). A taxonomy of robot deception and its benefits in hri. In *Proceedings of the ieee international conference on systems, man, and cybernetics* (p. 2328-2335).

Sjoberg, J., Zhang, Q., L.Ljung, Benveniste, A., Delyon, B., Glorenneca, P., ... Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, *31*, 1691-1724.

Srinivasan, M. (1995). Strategies for visual navigation, target detection and camouflage: inspirations from insect vision. In *Ieee international conference on neural networks* (Vol. 5, p. 2456-2460).

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. The MIT Press.