DYNAMIC GOAL COORDINATION IN PHYSICAL AGENTS

Jose Antonio Martin H.
Faculty of Computer Science, Universidad Complutense de Madrid
C. Prof. José García Santesmases, 28040, Cuidad Univesitaria, Madrid, Spain

Javier de Lope
Dept. Applied Intelligent Systems, Universidad Politécnica de Madrid
Campus Sur, Ctra. Valencia, km. 7, 28031 Madrid, Spain

Keywords: Dynamic Optimization, Goal Coordination, Robotics, Multi-Objective Optimization, Reinforcement Learning, Optimal Control.

Abstract: A general framework for the problem of coordination of multiple competing goals in dynamic environments for physical agents is presented. This approach to goal coordination is a novel tool to incorporate a deep coordination ability to pure reactive agents. The framework is based on the notion of multi-objective optimization. We propose a kind of "aggregating functions" formulation with the particularity that the aggregation is weighted by means of a *dynamic weighting unitary vector* $\overrightarrow{\omega}(S)$ which is dependant on the system dynamic state allowing the agent to dynamically coordinate the priorities of its single goals. This dynamic weighting unitary vector is represented as a set of $n-1$ angles. The dynamic coordination must be established by means of a mapping between the state of the agent's environment S to the set of angles $\Phi_i(S)$ using any sort of machine learning tool. In this work we investigate the use of Reinforcement Learning as a first approach to learn that mapping.

1 INTRODUCTION

The problem of designing controllers for physical agents (i.e. robots) is a very active area of research. Since its early beginnings in the middle of the past century many different approaches have been proposed. The control paradigms where changing gradually from the deliberative paradigm to the reactive paradigm and from pure mathematical and engineering formulation to more biological formulations, approximating very different fields of research. One of the fruitful branches of study has come to be called the bio-inspired paradigm or bio-mimetic approach (Passino, 2005).

In this paper we formulate a bio-inspired approach to solve the problem of multiple goal coordination for physical agents in dynamic environments. This problem is the task to assign the right priority to each agent's goal leading to the design controllers following a pure reactive approach. Pure reactive approaches have been applied successfully to tasks like robot navigation with collision avoidance. Despite that, there are more complicated tasks where a single reactive behavior is not enough. Such kind of tasks involve multiple goals where the agent needs to balance the importance of each particular goal. The most difficult problem from this point of view is when multiple goals are in conflict (competing goals), that is, by following a pure reactive behavior the direct attainment of one goal results in a loss in the attainment of another goal. This situation is very common in real tasks like collision avoidance vs. goal reaching and pursuit and evasion games.

In order to deal with this kind of problem still following a reactive bio-inspired approach we have developed a framework for the dynamic goal coordination problem that will allow the design of single reactive agents that are capable of solving complicated problems with conflicting and competing goals.

Usually a solution can be found by means of a combination of all the objectives into a single one using either an addition, multiplication or any other combination of arithmetical operations that we could devise. There are, however, obvious problems with this approach. The first is that we have to provide some accurate scalar information on the range of the objectives to avoid having one of them to dominate the others.

The approach of combining objectives into a single function, usually referred to as "aggregating func-

Antonio Martin H. J. and de Lope J. (2006).
DYNAMIC GOAL COORDINATION IN PHYSICAL AGENTS.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 154-159
DOI: 10.5220/0001216401540159

tions", and it has been attempted several times in the literature with relative success in problems in which the behavior of the objective functions is more or less well known. The problem of static multi-objective optimization has been actively studied by the evolutionary computation community (Fonseca and Fleming, 1995; Zitzler et al., 2002).

In this paper we propose a kind of "aggregating functions" formulation with the particularity that the aggregation is weighted by means of a *dynamic weighting unitary vector*, $\overrightarrow{W}(S) = (\omega_1(S), \ldots, \omega_n(S))$. By dynamic we mean that the unitary vector of weights is time varying or, more precisely, it depends on the system dynamic state allowing the agent to dynamically coordinate the priorities of it's single goals. This dynamic weighting unitary vector is represented as a $n-1$ set of angles: $\Phi_i(S) = (\phi_1(S), \ldots, \phi_{n-1}(S))$. The dynamic coordination must be established by means of a mapping between the state of the agent's environment S to the set of angles $\Phi_i(S)$.

2 GOAL COORDINATION

2.1 Goal Coordination in Static Environments

Goal coordination in static environments can be modeled as a Multi-Objective Optimization Problem (MOOP) where the objective function set corresponds one to one to the goal set in a natural way.

The standard form of a MOOP is presented in 1.

$$\min_{\mathbf{x}\in S}(J_1(\mathbf{x}), \ldots, J_n(\mathbf{x})) \tag{1}$$

where x is the design vector, S is the set of feasible solutions and n is the number of objectives.

One of the traditionals and very intuitive ways of solving a MOOP is to aggregate the objectives into a single scalar function and then minimize the aggregated function:

$$\min F(\mathbf{x}) = \sum_{i=1}^{n} J_i \tag{2}$$

where J_i represents each single objective function.

Also, assigning a weight to every objective is often used as a form to obtain more general solutions weighing this way the importance of each objective:

$$\min F(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot J_i(\mathbf{x}) \tag{3}$$

where $0 \leq w_i \leq 1$ and $\sum w_i = 1$.

The final extension to this approach comes from the idea that the weighting factors must by dynamic in order to deal with some complex forms of the Pareto Front.

$$\min F(\mathbf{x}) = \sum_{i=1}^{n} w_i(t) \cdot J_i(\mathbf{x}) \tag{4}$$

2.2 Goal Coordination in Dynamic Environments

In static optimization problems as we have seen, the solution or set of solutions are points in the *n-dimensional* space, instead in dynamic problems the solutions are trajectories or functions describing the varying optimal point due to the dynamic behavior of the objective functions.

Following the same approach of MOOP in static environments, we can model the goal coordination in dynamic environments as a Multi-Objective Dynamic Optimization Problem (MODOP).

The most frequent form of a MODOP is:

$$\min_{\mathbf{x}\in S}(J_1(\mathbf{x}, t), \ldots, J_n(\mathbf{x}, t)) \tag{5}$$

where x is the design vector, S is the set of feasible solutions , n is the number of objectives and t is the time variable which determines the dynamic behavior of the functions J_i. Now the functions J_i are dynamic in the time dimension.

A more general way of expressing a MODOP is as follows:

$$\min_{\mathbf{x}\in S}(J_1(\mathbf{x}, \Theta), \ldots, J_n(\mathbf{x}, \Theta)) \tag{6}$$

where x is the design vector, S is the set of feasible solutions, n is the number of objectives and Θ is a non direct controllable variable which determines the dynamic behavior of the functions J_i. Now the functions J_i are dynamic.

Note that Θ does not belong to the design vector $\mathbf{x}$, otherwise we will have a standard MOOP. It is the fact that Θ is not a design variable and hence a non direct controllable factor that transform the standard MOOP in a MODOP. The variable Θ is frequently expressed as the time variable t to indicate the time varying behavior of the functions although we prefer to express it in the general form.

Using an *endogenous* approach we can try to solve this kind of problem using the expression (4) but, due to the dynamic behavior of the objective functions, the solution is not a point in the *n-dimensional* space but a trajectory and in general a function of Θ for every design variable. Thus we can reformulate the problem as:

$$\min F(f(\Theta)) = \sum_{i=1}^{n} w_i \cdot J_i(f_i(\Theta), \Theta) \quad ; \forall\, \Theta \tag{7}$$

where $f_i(\Theta)$ is the design function set to be found, n is the number of objectives and Θ is a non direct

controllable variable which determines the dynamic behavior of the functions J_i.

Solving this kind of problem in general is hardly complicated, so we will restrict the problem to which are best suited to our purposes, that is, we will concentrate in the problem of goal coordination in dynamic environments for physical agents where a physical constraint holds.

The Physical Constraint Assumption:
Being $f_i(\Theta)$ the reaction of the physical agent then:

1. Each $f_i(\Theta)$ is *continuous* and *soft*.
2. Θ is *continuous* and *soft*.

This physical constraint reflects the fact that the physical agent have limitations in its reactions (i.e. limited velocity, acceleration, etc.) thus the position of the agent in the environment must describe a continuous trajectory making Θ *continuous* and *soft*. Formally, being (R_1, R_2) two consecutive reactions of a physical agent and (Θ_1, Θ_2) the respective posterior states then:

$$\lim_{||R_1 - R_2|| \to 0} ||\Theta_1 - \Theta_2|| = 0 \tag{8}$$

$$\lim_{||\Theta_1 - \Theta_2|| \to 0} ||f(\Theta_1) - f(\Theta_2)|| = 0 \tag{9}$$

In this work, we will deal only with environments which satisfy this physical constraint, indeed, the majority of problems in robot navigation, manipulator control and in general mechanical control problems satisfies this physical constraint.

2.3 A Bio-inspired Approach

A classical heuristic for dynamic environments which is indeed a bio-inspired approach to define the design function set subject to the physical constraint is to use the classical notion of negative feedback (sensory feedback) where the output of the system is the direction in which the change in the objective function is minimized with respect to the change in the control action as shown in (10).

$$x = -\mu \frac{\partial J}{\partial x} \tag{10}$$

where x is the control action, J is the objective to be minimized and μ is a proportional factor referred in the bio-inspired approach as the difference between the "intensity" of the stimulus and the proportional "intensity" of the reaction.

Using the sensory feedback control law, we can define the dynamic control variables in the next given form:

$$x_i(\Theta_t) = -\sum_{i=1}^{n} \mu_i \cdot \frac{\partial J_i}{\partial x_i(\Theta_{t-1})} \tag{11}$$

where Θ_t represent the state of the dynamical system and x_i, J_i and μ_i are been previously defined.

This approach is enough to solve some easy problems in dynamic single-objective problems, but in general this procedure alone can not work in dynamic multi-objective problems due to the obstruction in the competing objectives. In order to solve this situation, one solution is to add dynamic weighting factors to (11):

$$f_i(\Theta) = x_i(\Theta_t) = -\sum_{i=1}^{n} \mu_i \cdot \frac{\omega_i(\Theta) \cdot \partial J_i}{\partial x_i(\Theta_{t-1})} \tag{12}$$

where now, $\omega_i(\Theta)$ is the dynamic weighting factor which coordinates the priorities of the objective functions.

Thus we can define the general form of the system to be minimized as in (13) which is indeed equal to (7) without the w_i factor where w_i is now implicit in the functions $f_i(\Theta)$ as the $\omega_i(\Theta)$ factor.

$$\min F(f(\Theta)) = \sum_{i=1}^{n} J_i(f_i(\Theta), \Theta) \tag{13}$$

Once established the problem in this way the solution of the problem gets reduced to a problem of multiple goal coordination which is achieved by constructing a set of functions $\omega_i(\Theta)$. We will define each $\omega_i(\Theta)$ in the alternative form:

$$\sqrt{\sum_{i=1}^{n} \omega_i(\Theta)^2} = 1; \quad \text{where } -1 \leq \omega_i(\Theta) \leq 1 \tag{14}$$

By defining each $\omega_i(\Theta)$ in this form, the problem gets now reduced to finding the direction of the unitary vector $\overrightarrow{\omega}$ in the space whose axes are formed by the objectives J_i.

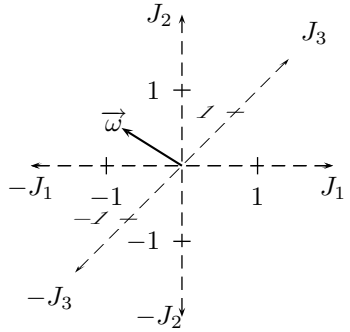

Figure 1: Coordinate frame and the $\overrightarrow{\omega}$ weigth vector.

The unitary vector $\overrightarrow{\omega}$ can be represented the set of angles ϕ_i and $i \in [1, 2, \ldots, n-1]$ which we can control the goal coordination. Fig. 1 shows a graphic representation of a coordinate frame for a system with three objectives.

Thus the problem is to construct the rotation function for all Θ. Note that with this definition (12) describes a continuous and soft function where the physical constraint holds.

It is important to show the clear difference and advantage of defining the dynamic weighting factors as shown in (14).

First, by defining the dynamic weighting factors in this way:

> it is possible to select a direction which is opposed to some particular objective J_i, this particular fact allows the control mechanism to escape from local minima and hence improve the goal coordination mechanism.

Second,

> we gain a degree of freedom because we need to control only $n-1$ angles to control the n dynamic weighting factors.

That is, in general, the set $\{\phi_1, \ldots, \phi_{n-1}\}$ can represent any direction for goal coordination which can not be achieved using the standard approach: $\sum \omega_i(\Theta) = 1$; where $0 \leq \omega_i(\Theta) \leq 1$ which only permits directions at least zero for all objective functions J_i.

3 EXPERIMENTS

We are experimenting with the dynamic goal coordination approach presented in continuous mobile agent tasks such as the problem of parking a car and a modified version of the mountain car problem.

Before presenting the experimental results we will describe the framework that we are using.

3.1 Experimental Framework

1. The first step towards this approach is to clearly define the objective functions or performance indexes that will be minimized that is, the goals that the agent must reach:

$$\text{Goals} = (J_1, \ldots, J_n)$$

2. The set of state variables must be clearly defined we will denote the set of state variables as:

$S = (s_1, \ldots, s_n)$ where each s_i is continuous

3. Define the set of control variables:

$X = (x_1, \ldots, x_n)$ where each x_i is continuous

4. Finally, the unitary vector whose rectangular components are the set of weighting functions must be described:

$$\overrightarrow{W}(S) = (\omega_1(S), \ldots, \omega_n(S))$$

Thus, we use the set of angles $\Phi_i(S) = (\phi_1(S), \ldots, \phi_{n-1}(S))$ to construct the unitary vector $\overrightarrow{W}(S)$ with the formulation shown in (15).

$$\begin{aligned}
\omega_1 &= \sin(\phi_1)\sin(\phi_2)\sin(\phi_3)...\sin(\phi_{n-1}) \\
\omega_2 &= \sin(\phi_1)\sin(\phi_2)\sin(\phi_3)...\cos(\phi_{n-1}) \\
\omega_3 &= \sin(\phi_1)\sin(\phi_2)\sin(\phi_3)...\cos(\phi_{n-2}) \\
\omega_4 &= \sin(\phi_1)\sin(\phi_2)\sin(\phi_3)...\cos(\phi_{n-3}) \\
\vdots &= \qquad\qquad \vdots \\
\omega_n &= \cos(\phi_{n-(n-1)}) = \cos(\phi_1)
\end{aligned} \tag{15}$$

Finally, the set:

$$\mathcal{M} = (\mu_1, \ldots, \mu_n)$$

must be carefully adjusted by means of some optimization technique or by providing direct information by the system engineer.

Therefore, the control law in (16) gets completely defined:

$$x_i(S_t) = -\sum_{i=1}^{n} \mu_i \cdot \frac{\omega_i(S_t) \cdot \partial J_i}{\partial x_i(S_{t-1})} \tag{16}$$

3.1.1 Learning Goal Coordination

In order to learn the goal coordination function:

$$\Phi_i(S) = (\phi_1(S), \ldots, \phi_{n-1}(S))$$

we investigate the use of Reinforcement Learning as a way to learn the mapping from sates to angles, that is:

$$S \mapsto \Phi_i = (\phi_1, \ldots, \phi_{n-1})$$

where S is the state of the system (i.e. sensor readings, perception, etc.) and Φ_i is the set of angles which represent the unitary vector $\overrightarrow{W}$ for weighting each goal J_i.

We have selected the SARSA algorithm (Sutton and Barto, 1998) for our experiments. In order to discretize the state variables we are using CMAC tiles (Albus, 1975; Sutton, 1996) to discretize actions we elaborate an array of angles, so each action numbered from $1..n$ selected by the algorithm represents an array index where the real values of the angles are selected.

3.2 The Car Parking Problem

The Car Parking problem consists of conducting a car which is a non holonomic robot to a parking location with coordinates (x_f, y_f) and an orientation α_f. This problem is formulated as a differential game in (Isaacs, 1999).

3.2.1 System Design

Single objective goals of the problem can be defined as:

$$\begin{aligned} J_1 &= \frac{1}{2}(x_f - x)^2 \\ J_2 &= \frac{1}{2}(y_f - y)^2 \\ J_3 &= \frac{1}{2}(\alpha_f - \alpha)^2 \end{aligned} \quad (17)$$

where x, y are the coordinates of car, and α is the orientation of the car. The coordinates $(x, y, \alpha, x_f, y_f, \alpha_f)$ are taken from an external absolute reference system.

The state set $S = (s_1, s_2, s_3)$ is defined as:

$$\begin{aligned} s_1 &= (x_f - x) \\ s_2 &= (y_f - y) \\ s_3 &= (\alpha_f - \alpha) \end{aligned} \quad (18)$$

Finally the control variable θ is the angle to be applied to the car wheels which will follow the control law:

$$\theta(S_t) = -\sum_{i=1}^{n} \mu_i \cdot \frac{\omega_i(S_t) \cdot \partial J_i}{\partial \theta(S_{t-1})} \quad (19)$$

3.2.2 Experimental Setup

The experiment was made with the car starting point at $(x, y) = (0, 0)$ and a fixed starting angle of $\alpha = \pi/2$. The parking final position and orientation were selected randomly in an area determined by a circunference centered at the car initial position. A very important factor in Reinforcement Learning experiments is the selection of the reward function as well as the function that determines if the car has reached the goal. Although in the majority of the examples in Reinforcement Learning the reward function uses only $(-1, 0, +1)$ values, we are using a continuous reward function which is an inverse function of the summation of the performance indexes meaning that high rewards corresponds to minimums errors in the performance indexes.

$$reward = \frac{1}{1 + \sum J_i} \quad (20)$$

3.2.3 Results

The simulations where made using the Reinforcement Learning Framework (Sutton, 2006). In the Car Parking problem, experiments where conducted for multiple parking locations and orientations, the results indicate that the agent following the presented goal coordination approach learns a good enough policy for reaching the goal but not the optimal one, since some failed attempts to reach some goal configurations.

3.3 The Conflicting Mountain Car Problem

The traditional mountain car problem is the task to learn a policy or controller that leads to a car to the top of a mountain. Initially the car is in a valley between two mountains, the difficulty is that the car speed is not sufficient to overcome the force of the gravity, hence the car will have to use its inertia to reach the top of the mountain by a balancing strategy. This problem was solved in (Sutton and Barto, 1998) pp. 214, using the SARSA algorithm.

Here we present a variation of the problem in which the car goal is not only to reach the top of the mountain but also to reach the top at 0 velocity. Note that with this simple modification there are to obstructive goals due to for reaching the top of the mountain the car must accelerate in its direction but for reaching 0 velocity the car must decelerate in the opposite direction.

A representation of The Mountain Car Problem is shown in Fig. 2.

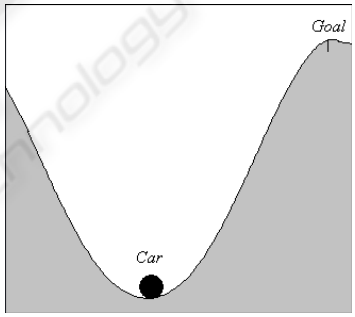

Figure 2: Car mountain problem.

3.3.1 System Design

Single objective goals of the problem can be formulated as:

$$\begin{aligned} J_1 &= \frac{1}{2}(x_f - x)^2 \\ J_2 &= \frac{1}{2}(v_f - v)^2 \end{aligned}$$

where x, v are the current position and velocity of the car car,and x_f, v_f are respectively the desired final position and velocity of the car. Note that J_1, J_2 are in this case conflicting objectives.

The state set $S = (s_1, s_2)$ is defined as:

$$\begin{aligned} s_1 &= (x_f - x) \\ s_2 &= (v_f - v) \end{aligned} \quad (21)$$

Finally the control variable θ is the throttle to be applied to the car which will follow the control law:

$$\theta(S_t) = -\sum_{i=1}^{n} \mu_i \cdot \frac{\omega_i(S_t) \cdot \partial J_i}{\partial \theta(S_{t-1})} \qquad (22)$$

3.3.2 Experimental Setup

Although in the majority of the examples in Reinforcement Learning the reward function uses only $(-1, 0, +1)$ values, we are using a continuous reward function which is an inverse function of the summation of the performance indexes meaning that high rewards corresponds to minimums errors in the performance indexes.

$$reward = \frac{1}{1 + \sum J_i} \qquad (23)$$

3.3.3 Results

The simulations where made using the Reinforcement Learning Framework (Sutton, 2006). In the Conflicting Mountain Car Problem, the results indicate that the agent following the presented goal coordination approach learns a near optimal policy for reaching the goal by successive approximations.

Figure 3, shows the result of the simulation, showing the fact that the number of steps for reaching the goal decrease up to a constant number of steps meaning that this is the near optimal strategy for controlling the agent.

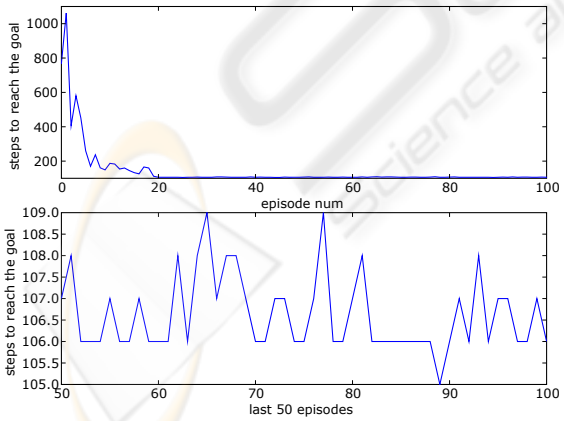

Figure 3: Mountain Car experimental results.

In Figure 3(Top) it can be seen that the number of steps to reach the goal in the firsts episodes is near 1000 and in the subsequent episodes this step number decreases considerably, indeed, Figure 3(Bottom) shows that the systems arrives to a stable configuration around 107 steps per episode. An additional observation is that the speed of learning is high, from the episode number 20, the system is practically stabilized around the 107 steps per episode.

4 CONCLUSIONS AND FURTHER WORK

A general framework for the problem of coordination of multiple competing goals in dynamic environments for physical agents has been presented. This approach to goal coordination is a novel tool to incorporate a deep coordination ability to pure reactive agents.

This framework was tested on two test problems obtaining satisfactory results. Future experiments are planed for a wide range of problems including differential games, humanoid robotics and modular robots.

Also we are interested in the study of methods for reinforcement learning better suited for continuous input states and multiple continuous actions. Usually a discretization of input and output variables is produced but in other cases a better result could be approximated if the problem were modeled by means continuous states, generating more robust systems.

REFERENCES

Albus, J. (1975). A new approach to manipulator control: The cerebellar model articulation controller (cmac). *J. of Dynamic Sys., Meas. and Control*, pages 220–227.

Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16.

Isaacs, R. (1999). *Differential Games*. Dover Publications.

Passino, K. (2005). *Biomimicry for Optimization, Control, and Automation*. Springer Verlag.

Sutton, R. (2006). Reinforcement learning and artificial intelligence. http://rlai.cs.ualberta.ca/RLAI/rlai.html.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning, An Introduction*. MIT Press.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Touretzky, D. S., editor, *Adv. in Neural Inf. Proc. Systems*, volume 8, pages 1038–1044. MIT Press.

Zitzler, E., Laumanns, M., Thiele, L., and Fonseca, C. (2002). Why quality assessment of multiobjective optimizers is difficult. In *Proc. GECCO 2002*, pages 666–674.