A New Dynamic Model for a Multi-Agent Formation

Daniel Gonçalves[1], Pedro Machado[1], and Ana Paiva[2]

[1] Instituto Superior Técnico, Technical University of Lisbon
[2] Instituto Superior Técnico, Technical Univesity of Lisbon and INESC
(djvg@gia.ist.utl.pt, pedro.machado@ip.pt, Ana.Paiva@inesc.pt)

Abstract. Many systems have been already developed concerning agent teams in a world with obstacles. One of the problems of such systems lies on how to maintain a pre-defined formation when we have several agents moving in the world.In this paper we defend that, in order to have a robust and realistic system, a control model that includes the notions of mass and acceleration must be used. To prove that, we developed a control system based on the classic mechanical physics, which is a force-based model. From the results obtained we can see that, although some problems arise when using such realistic kind of model, they are solvable and the quality of the simulations performed by the system is significantly better than the simulations obtained using other control models.

Keywords: Multi-Agent Systems, Multi-Agent Formation

1 Introduction

Amongst several known problems in the field of mobile robotics, the most usually discussed is how to allow the robot to move autonomously in an unknown environment. This problem becomes worse when we consider not only a single robot, but a team of them given that a robot must consider not only the terrain topology, but also the positions and movements of the remaining robots, to prevent collisions and to stay out of their way. To handle such problem we must therefore establish a *group behavior* that allows the robots to perform in the right way.

Recently, these problems have been the object of study of the area of *software agents* [5] and multi-agent systems. One of the distinguishing features of this area is the fact that each agent is responsible for determining its own actions (resorting only to it's knowledge of the world, given by the sensors), not being controlled by any external process. This tries to mimic what we can find in nature, where no higher intelligence determines the individual actions. Rather, the overall behavior of the society of agents emerges from the individual decisions. The software agent's approach seems to be not only more adequate, but also more correct to deal with this problem.

In this area, several systems have already been developed. The most famous of these is undoubtedly, the *BOIDS* system, created by Craig Reynolds [3]. Based on the same approach other systems have been developed, such as, for example the system by Hodgins and Brogan [2] that simulates a herd of pogo-stick-like robots in a tri-dimensional world. One aspect of these systems is that they do not maintain any kind of formation or differentiate the agents and specify their desired positions in relation to each other. However, formations are important since they allow the team to use its sensory assets in a more efficient way than if the team was arranged randomly [1]. This paper focuses exactly in the formation control of a team of robots.

Moreover, we want for the formation to remain robust enough in the presence of unpredicted obstacles. One system that achieves these goals was created by Balch and Arkin [1]. It is based on a small number of robots able to maintain a pre-determined formation while moving towards a goal, even in the presence of obstacles. However, some limitations can be found on their approach, in particular its lack of realism.

In this paper we defend that such kind of system should have a more realistic dynamic model, thus, including the notions of mass and acceleration. The work here described introduces these notions through the use of a formation control system based on the classic mechanical physics, thus, a force-based model. We will show the problems that arise when using that realistic kind of model, and how we solved them.

This paper is organised as follows. In the next section, we will discuss the problem domain and the entities involved in it. Next, we will rapidly describe the main characteristics of Bach and Arkin's system [1], and explain the reasons behind the need for some improvements. Then, we will show what makes our system different from the existing ones and the results we achieved with it. Finally we will discuss the results and point out the conclusions our work led to.

2 The Problem Domain

The system created simulates a society where several agents can move in a world with obstacles, whilst trying to reach a goal position. Since this system can be seen as an extension of the original work by Balch and Arkin [1] we introduce the basic concepts involved very briefly.

2.1 The Formations

In order to make possible the definition of a formation, it is necessary to distinguish an agent from its companions. Thus, we attribute an unique number to each agent. A very large number of formations are, of course, possible. There are, however, four standard formations in military domains, depicted in Fig. 1:

Line: the robots travel side-to-side.
Column: the robots travel behind each other.
Diamond: the position of the robots is that of the vertexes of a diamond.
Wedge: the robots are positioned in a "V" shape.

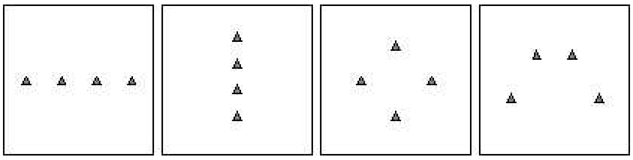

Fig. 1. Possible formations

These formations, besides allowing the verification of the validity of the system in that domain, are rich enough to test a great amount of situations in simulation terms, each one having its own problems, as we will see below.

2.2 Reference Methods

There are three classical reference methods: in *relation to a given robot* , in *relation to a leader*, and in *relation to the formation's center of mass.* The first method is similar to the one used in BOIDS [3], with the difference that in BOIDS the position of a bird was dependent on those of its neighbors, which could be any other birds. In here, the robot from which the position is determined is given from the start. The definition of the desired position of a robot in relation to its reference point is given in terms of two values: an *angle* and a *distance.*

The angle must be measured not in relation to the horizontal, but in relation to the perpendicular of the movement of the reference point. This will allow the formation to maintain itself when not moving straightforward.

It's velocity vector, in the case of a single robot gives the direction of the movement of the reference point. In the case the reference point is the center of mass of the formation, the average of the individual velocity vectors is used.

2.3 The World

The world is a bi-dimensional square with arbitrary size, where several *obstacles* (columns with a given radius) can be found.

Apart from the obstacles, the only other important points in the world are the *goals.* A goal is a point that represents the place the robots must go in order

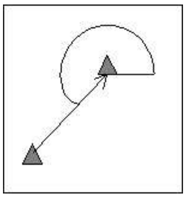

Fig. 2. Relative reference

Table 1. Motor Schemas

Motor schemas	**Usefulness**
Avoid-Static-Obstacle	Avoid collisions with static obstacles
Avoid-Robot	Avoid collisions with other robots
Move-To-Goal	Drive the robots towards the goals
Noise	Noise
Maintain-Formation	Force the robots to maintain the formation

to fulfil their mission. In the case there are several goals in the world, the robots must get to them in a pre-specified order. The robots are considered to have achieved a goal when the center of mass of the formation is less than a given number of units from the goal.

2.4 The Motor-Schemas

The formation behaviors of the robots were implemented as motor-schemas (Table 1). These schemas are similar to the ones used by Balch and Arkin.

From each schema we have a vector (usually a force), ranging from zero to a preset maximum. All the resulting vectors are added, taking into account their relative gain. The resulting vector that determines, at each instant, how will the robot's movement be altered.

Avoid-Static-Obstacle and Avoid-Robot: these schemas' functions are used to prevent collisions with an obstacle or with other robots. The resulting vector will be in line with the line that joins the robot and the obstacle and the direction that will keep them both apart. The intensity of the vector will depend on two values: a *minimum* and a *maximum range*. If the distance between the robot and the obstacle is grater that the maximum range, then the intensity of the vector will be zero. If it is smaller than the minimum range, it will be the maximum permitted intensity. Otherwise, it will oscillate between those two values.

Move-To-Goal: this schema is the responsible to make a robot move in direction of it's objective. Its result is simply a vector with the maximum intensity and the direction of the goal.

Noise: to make the simulation more realistic, this schema results in a vector with random direction and intensity, thus introducing noise in the system.

Maintain-Formation: this is, perhaps, the most important schema, since it allows the robots to position themselves in the desired formation positions. First, the desired position is determined, in relation to the established reference point. Then according to the distance of the robot to that point, the intensity of the vector with its direction will vary.

Like in the avoidance schemas, two values are considered: a minimum and a maximum radius. In this case, however, if the distance to the desired point is greater than the maximum range, the intensity of the vector will be the greatest allowed. The desired point is then said to be in the *"ballistic zone"*. If the distance is smaller than the minimum radius, the intensity of the vector will be zero (*"dead zone"*). Otherwise, the desired point is in the *"controlled zone"* and the vector's intensity is proportional to the distance.

2.5 Limitations of the Existing Approach

The mechanisms we described until now were used in system developed by Balch and Arkin [1]. However, there are some limitations to their work. Firstly, the reference method that gives the desired position of a given robot in relation to that of another robot was not implemented. Secondly, and by far the most important omission of this work is that the simulation model is based only on the velocity, and not acceleration. This means that the vectors returned by the motor-schemas are, in fact, an indication of the desired velocity in the next instant. This makes the simulation very unrealistic, since it assumes there can exist infinite accelerations, that can radically change the speed vector of a body instantly. It also simplifies greatly the control problems, making the task of maintaining the formation a very easy one.

These limitations motivated the creation of our system, which will be described and evaluated in the following sections.

3 A Realistic Simulation Model

To overcome the limitations presented, a new system was created. Mainly, we have improved the existing system on three factors: new types of obstacles, new vector intensity decay models and a dynamic force-based control model.

3.1 New Types of Obstacles

In the real world, it is very unrealistic to assume that all the obstacles an agent will face are fixed in nature. Not only that but it is plausible that the agent would react differently when faced with different obstacles. It might, for instance, start to deviate from larger ones first (since they are circular in nature, they will occupy a greater area to avoid, requiring a larger deviation).

So, in the first place, we assumed our agents have some kind of "improved sensor" that can not only detect what is the distance to the nearer obstacles, but also the *curvature radius* of them. Our *avoid-static-obstacle* motor-schema will take this information into consideration, and the resulting vector will be proportional to the size of the obstacles.

Besides this change to the static obstacles, we also introduced another kind of obstacles: the *mobile obstacles.* These are other agents that, unlike the robots, are limited to roam the landscape in a pre-determined or random way, thus hampering the voyage of the robots. We created these obstacles in order to account for the multitude of unexpected features the robots might encounter on a real situation (people, animals, and so on). To deal with these obstacles, a different schema, *avoid-moving-obstacle*, was introduced.

3.2 Vector Intensity Decay Models

When determining what will be the intensity of the vector returned by a motor-schema, two radiuses are normally used. In the case of the *avoid-static-obstacle* scheme, for instance, when the distance between the obstacle and the robot is superior to a given radius, the intensity will be zero. If it is smaller than a certain value, the intensity will be the maximum allowed. Between those two values, the intensity will vary. In the original system, the variation of the intensity of that vector was *linear*. In our model, we have implemented a *quadratic* decay.

We found that often, this aspect added an "urgency feeling" to the robots, as the repulsion (or attraction, depending on the schema) will increase greatly in extreme situations, but will be moderate otherwise. Since we used an acceleration model, if, by any chance the robots get into one of those extremes, the response must be swift. In other cases, we do not want the robot to use very large accelerations since it would make the task of controlling it more difficult.

3.3 The Dynamic Model

In our system, we have implemented a control model where the vectors resulting from the motor-schemas are, in fact, force vectors. We call this control model the *acceleration model*. According to that mass of the robot, we generate an acceleration vector (up to a given maximum), that we will use to alter the movement of the robot at the next instant of time.

We also introduced *attrition* into the system. This was done by defining an attrition constant that generates a force with a direction opposite to that of movement in each time step. This makes it very difficult for the robots to saturate the system by getting to the *maximum speed*. This speed is still possible, but only temporarily and due to a very large acceleration which, as we have seen, occurs in extreme cases, using the geometric decay model. Thus, a robot falling behind it's desired position will, it the situation gets too bad, give an "extra push" that will be enough to compensate temporarily for the attrition and get in formation.

Likewise, even when travelling a long time in a straight line, the speed will not be at it's maximum, thus allowing for a correct change of direction.

All these problems, while relevant when simulating a real environment, do not occur in the velocity model (where the output of the schemas is a velocity vector, rather than a force).

3.4 Problems Introduced by the New Dynamic Model

The force-based dynamic model brought some new problems to the system. In the following paragraphs, we will briefly discuss them and analyze possible ways to solve them. This was done by carefully tuning a series of parameters that define how the system will behave.

- **Global Constants:** the value of the *maximum acceleration* had to be chosen very carefully. A small value would not be enough to permit the robots to react quickly enough to a sudden need of change in direction (such as when taking a curve). Likewise, a very large *maximum speed* would result in a similar behaviour. Thus, it was necessary to choose these two values correctly in relation to each other, in order to have accelerations large enough to account for the speed of the robots, but not large enough to make them loose control at the slightest change in movement. This was related to the attrition constant that we chose (that ended up being 0.1).
- **Maintain-Formation Parameters:** although this might not be evident, to make the force returned by this schema large in relation to that of the others (or to the maximum acceleration), might have ill effects.
  For example, if when getting out of formation, a robot heads with great speed towards the desired position, once it reaches that position, it would not be able do decelerate fast enough to stay there. So, we will witness an oscillatory behaviour, where the robot will first move very fast towards the position in one direction, and then in the other (Fig. 3).
- **Move-To-Goal Parameters:** the *move-to-goal* schema might not be so easy to define as it would seem. If the force produced by this schema is very large, the robots will try to head towards the goal no matter what lies in their path. This will hamper greatly the robots' capability to avoid obstacles.
  Also, if the acceleration produced by this schema is very high, it will be difficult for the robots to change direction once they have reached a goal, towards another one. So, we must be very careful while choosing the value of the gain for thus schema.
- **Maintain-Formation vs. Move-to-Goal:** another important trade-off was between the *maintain-formation* and the *move-to-goal* schemas. A situation that usually arises is that the robots, when travelling in a straight line towards a goal, tend to narrow the formation.
  This occurs because the forces produced by the *move-to-goal* schema all point to the same place, and do not take into account that the actual "goal" to each robot is different according to it's position in the formation. Because the robots will travel for some time along a straight line, they will have time to greatly increase their speed towards the goal. So, if the *maintain-formation* force is not large enough, it will not suffice to avoid the narrowing effect.

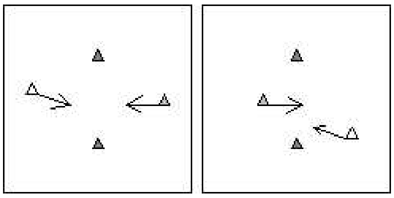

Fig. 3. Oscillatory behaviour

Table 2. Default parameters

Parameter	Value	Parameter	Value
Avoid-Static-Obstacle		Avoid-Moving-Obstacle	
Gain	3	Gain	1.5
Maximum Range	50	Maximum Range	50
Minimum Range	5	Minimum Range	5
Avoid-Robot		Maintain-Formation	
Gain	2.5	Gain	4
Maximum Range	10	Maximum Range	40
Minimum Range	5	Minimum Range	3
Move-To-Goal		Noise	
Gain	3	Gain	0.05

Avoidance Schemas: we must take care with the forces resulting from these schemas, in order to allow the robots to attain a goal that lies behind an obstacle (if the repulsion is too large, they will never be able to pass beyond that point). We can adjust these schemas in two ways: with a large intensity on a small radius and vice-versa. With a small radius, the influence of the schema will be felt independently by each robot (since some of them might be inside the radius and others not). With a large radius, the influence of the schema is felt more uniformly on the entire formation.

Finally, the *avoid-robots* schema must only yield a force in extreme situations, since, otherwise, we risk losing control with the antagonism between this schema and *maintain-formation*.

4 Tests

We conducted some tests with the same velocity model as Balch and Arkin [1] and the results we got were similar. To perform the tests with the acceleration model, and taking into account the previous considerations, some values for the parameters were chosen (see Table 2).

The robots' maximum velocity was limited to 30 spatial units/time unit and they were given the mass of one. The robots' dimension varies in unrelated tests. Towards the objective of making the *avoid-robot* schema an emergency force, we gave it a gain of 2.5, an influence sphere of 15 and a minimum radius of 5.

In all tests, the world is similar, with a dimension of 1000x1000 spatial units and no mobile obstacles (to permit comparing the tests). The sample period was 0.1 time units and the value of noise gain was 10% of the maximum acceleration.

4.1 Statistical Evaluation

The main evaluation of the system is qualitative. Despite that, we tried to add some objective evaluation and some more concrete manner to compare the performance of the system with altered parameters or in different situations. Therefore we considered:

Path Ratio. This value is calculated dividing the robots average traveled space by the distance between the various goals. This gives a notion of the deviation of the robots to the correct path. Notice that this ratio has a minimum of 1, and is normal in the simulations to have values substantial greater than this limit - some of the robots, in order to *maintain-formation*, have to perform same kind of external path regarding the line between goals;

Formation Error. Is the sum of the position error of each of the robots towards is ideal position in every instant of the simulation. This contains information about the formation maintenance;

Average Formation Error. Is the average of the above value;

Simulation Time. Length of the simulation, in time units or number of samples.

When defining a robots position in the formation in relation to another robot it is important, in order to achieve good results, to consider carefully what robots should be related. The immediate option, defining one robot in relation to is nearest neighbor is not always the best solution.

4.2 Results Obtained

1. Column Formation A formation with 50 spatial units between the robots and a robot dimension of 10 was used. When using the definition of the position in relation to center of mass, the behavior, in qualitative terms, of the formation is good. The average formation error is low (36). Note that a very small change in the orientation of the reference point, either a robot or, in this particular case, the formations center of mass, causes an enormous variation of the ideal position of the robot. The path ratio is around 1.07. This is explained by the tendency in performing the curves external to the rectilinear path between goals.

In case of formation relative to a leader, the error values are considerably larger (91.0 in the average and 421.2 to the maximum). This is because in this case the reference point (the leader) has quicker oscillations than the center of mass, introducing a non real error measure (the maximum error happen when in the end of a rectilinear path the leader suddenly change is orientation).

When facing an obstacles field, the robots in this formation, independently of it's definition, tend to follow the same path, like a snake.

2. Line Formation A formation with 50 spatial units between the robots and a robot dimension of 8 was used.

This formation, when faced with a obstacle, produces a position error larger than the column formation. Take in account that in order to avoid an obstacle a robot must change it's movement perpendicularly to its actual direction.

In the case of a relative formation, the gain of the *maintain-formation* schema must be very small (0.4) to prevent an increasing oscillation, that, in ultimately

Table 3. Test results of the different formations

	Column		**Line**		**Diamond**			**Wedge**	
	Leader	CoM	Relative	CoM	Relative	Leader	CoM	Relative	CoM
Path Ratio	1.08	1.07	1.10	1.11	1.14	1.08	1.07	1.08	1.06
Formation Error									
Minimum	17.41	6.43	1	2	10	0	1	13	2
Average	91	36.3	583.3	152.3	200	166	72	341	225
Maximum	421.2	106.4	1790	427.2	570	450	251	836	839
Std. Dev.	70.6	25.7	435.5	96.7	121	124	50	214	181
Time	2581	1765	2198	3086	3087	2648	2693	2839	3216

makes the robots go around each other. We didn't find a set of value parameters that made the performance of this kind formation definition acceptable.

As an important note on the behavior of this formation is the narrowing that it suffers when approaching a goal.

3. Diamond and Wedge Formations In the diamond formation, the distance between the robots and the center of mass was fixed to 75 units, and the robots dimension to 10. The increase of this distance forced a relaxation of the goal achieving condition. Thus, the desired distance to a goal was incremented to 40.

The best results, with the formation in relation to a leader or in relation to the center of mass, were achieved with an avoid obstacles schema concentrated and powerful (with a gain of 9 gain and a radius of 20). On the other hand, in formation relative to another robot, the best results were obtained with a long range schema and low gain (100 and 2 respectively).

Note that, in the formation defined in relation to a leader, if the leader robot turns, the others maybe go back a little or change their path in order to maintain the formation. Because the leader has no concern about the formation, he has a tendency to get to a goal faster than the others, and then have to wait for them.

On the Wedge formation, the results of the tests are similar.

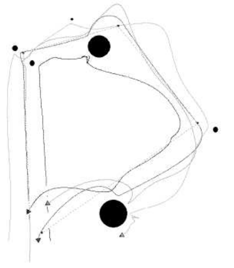

Fig. 4. Diamond formation defined in relation to a leader

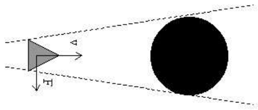

Fig. 5. Ideal avoid obstacle force.

5 Discussion

The results we found lead to some questions, both empirical and theoretical which will be discussed bellow.

5.1 Schemas parameters

In all the cases, we encountered that a low radius but a strong gain of the *avoid-robots* schema produces the best results. In fact, this emergency force, in a trivial situation, is unnecessary: the *maintain-formation* schema will produce the desired repulsive force when the robots are approaching each other.

When concentrated, the *avoid obstacle* schemas produce a behavior where each robot travels along the obstacles' borders. Some times there is a transformation in this normal, and observed, behavior and the robots try to avoid the obstacles in an oscillatory manner - approaching and retreating quickly.

A particular situation occurs when a robot is following a path that crosses an obstacle's center. The force generated by this schema will only slow down the robot's velocity, but it will not induce a shift strong enough to make the robot contour the obstacle. Also, this force is maintained when the robot passes the obstacle, now with the form of a positive acceleration, like if the robot is running away from the obstacle.

One improvement to this schema, would be to make the force have an orientation somewhat different from the current. In the case of Fig. 5, for example, the ideal force was perpendicular to the movement.

One last observation: this schema's gain must be substantially larger than the *move-to-goal* schema gain, in order to prevent a deadlock situation.

When a robot achieves a goal it continues to suffer the effects of *move-to-goal* schema, until all the formation achieves that goal. This causes, particularly in leader referenced formations, where the leader is in the front, that the robot keeps an oscillatory movement around the goal. It's suggested as future work to keep the force of this schema as zero in the neighborhood of the goal.

This last schema is also responsible for the narrowing, and slow reaction in emergency situations. To restrain its influence to the essential, we suggest that this schema's gain should be proportionally inverse to the velocity of the robot.

5.2 Formation Definitions

In every formation, the narrowing increases proportionally to the space between robots. This introduces large error values. The relative defined formations work better when the *maintain-formation* schema uses a larger radius to limit the controlled zone. This type of formation has big error values, even if the formation is well formed (look at the minimum error values for these formations).

In a leader-defined formation, non-leader robots should have zero gain in the *move-to-goal* schema, thus, they should be limited to following the leader.

5.3 Statistical Evaluation

The *formation error* used is an ambiguous statistic, and should be used with great caution. For example, it can be a good method to compare different tests with the same formation, but it's not so good between different formation definitions. In order to quantify the oscillation that robots suffer, the energy spent in a simulation must be considered in a future work.

6 Conclusions

As shown by the performed tests, by using the new simulation model more realistic results are obtained. Indeed, the old model based only in the velocity of the agents is not only unrealistic, but also much more sensitive to noise, thus introducing more oscillations in the robots. It also, requires a careful-tuning of the schemas to prevent unlimited velocities. Aiming to create a system that can eventually be used in a real-world environment, a velocity model is evidently inadequate. When presented to different situations (different formations, for instance) it doesn't have the robustness we would desire, needing to be tuned differently for each case.

The acceleration model we have presented in this paper brings new advantages to the simulation but it also introduces some complexity in the control system. It is not evident what schemas should be considered. Also, the tuning of the several parameters is not evident. However, once it is tuned, it performs correctly in a wide range of situations as shown by the results presented.

References

1. T. Balch and C. Arkin. Motor schema-based formation control for multiagent robot teams. In *Proceedings ICMAS-95*, pages 10–16, 1995.
2. J. Hodgins and D. Brogan. Robot herds: Group behaviours for systems with significant dynamics. In *Proceedings Artificial Live IV*, pages 319–324. MIT Press, 1994.
3. C. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 4(21):25–34, 1987.
4. C. Reynolds. Not bumping into things. notes on 'obstacle avoidance' for the course on physically based modeling at siggraph'88. Technical report, 1988.
5. Russel and Norvig. Intelligent agents. In *Artificial Intelligente: A Modern Approach*, pages 31–52, 1994.
6. Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings Artificial Live IV*. MIT Press, 1994.