COLLISION DETECTION AND PREDICTION USING A MUTUAL CONFIGURATION STATE APPROACH

Albert Schoute

University of Twente
Centre for Telematics and Information Technology (CTIT)
P.O. Box 217
7500 AE Enschede, Netherlands

ABSTRACT

A configuration state approach is presented that simplifies the mutual collision analysis of objects with known shapes that move along known paths. Accurate and fast prediction of contact situations in games such as robot soccer enables improved anticipatory and corrective actions of the state estimation, control and planning processes. An overview is given how advantage is taken of the configuration state approach in order to deal with collision state correction and collision avoidance in the robot soccer control system MI20 developed at our university.

1. INTRODUCTION

When playing a ball game some types of "contact" between the ball, the players and the environment are allowed by the rules, some are illegal but tolerated and some are strictly forbidden. Of the contact types allowed some will be "productive" and may be exploited, others are "counter-productive" and are better avoided.

In the case of a computer controlled game like robot soccer, accurate detection and prediction of contact situations, in this paper termed "collisions", can be advantageous in many respects. First, proper recognition of collisions between objects (ball, players and for example the field border) can support the state estimation process. Second, player control process(es) can take advantage of near collision information, in case it is useful to engage in the contact situation as well as when it is desired to avoid contact. And third, planning processes can foresee the occurrence of collisions in their plans and adapt these.

In this paper an approach is presented to simplify the determination of collision states of objects with known shapes on known paths. As collisions are mutual events between objects, any joint configuration state of two independently moving objects can be determined as a "colliding" state yes or no. Areas of collision states in the joint configuration space may be identified and characterized analytically for simple geometric cases. As two moving objects are represented by a motion trajectory in the joint configuration space, forthcoming collisions states may be predicted.

We will give an overview how the configuration state approach is applied in the robot control system MI20 [2].

2. MUTUAL COLLISION ANALYSIS

Consider two vehicles V_i (i=1,2) each traveling along a path with s_i being the current 1-dimensional position measured along the path. The joint state of the vehicles is represented by a point (s_1, s_2) in a 2-dimensional configuration space[1]. This concept is visualized in Figure 1. A configuration state is a collision state if the shapes of V_1 and V_2 overlap (Figure 2).

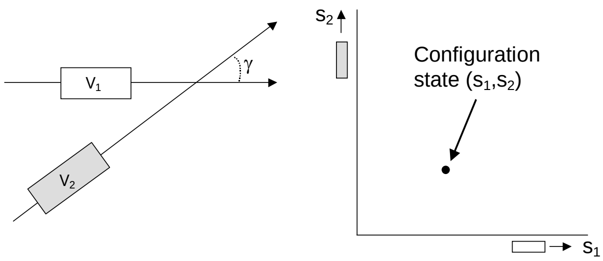

Figure 1. Mutual configuration state approach: the joint vehicle state is "pinpointed" as a point in configuration space by the coordinates (s1, s2).

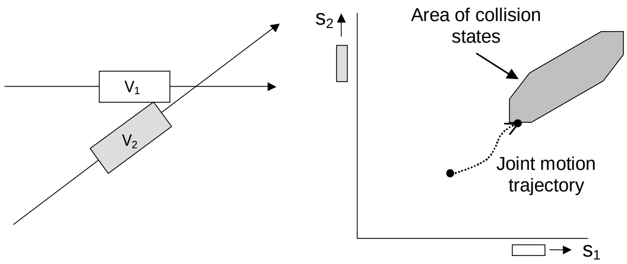

Figure 2. Joint motion leading to a collision state.

---

[1] Configuration space (C-space) is a key concept for motion planning in the area of robotics. See [1].

For vehicles with simple shapes traveling along straight paths, the collision state area can be characterized analytically. The form of the area solely depends on the shapes of vehicles and the intersection angle γ of the paths. Examples of collision areas are depicted in Figure 3.

If object shapes and/or paths become more complex collision areas can be computed only in an approximate manner. The RoadPlan analysis tool [5] can handle paths composed of straight line and circle sections, together with vehicle shapes that are round, rectangular or a mixed combination. The RoadPlan program uses a quadtree decomposition technique [1] to decide whether cells of configuration space are (non-)collision areas. An example of the collision analysis as presented by Roadplan is shown in Figure 4.

3. COLLISION DETECTION

In contrast to mathematical collision modelling based on the abstract concept of "overlapping" objects, collision detection in practice is not so straightforward. Stating the actual occurrence of a collision in real life is a matter of circumstantial evidence. In case of physical side effects such as damage or sound, collisions are easily recognized. But what if we have visual state information only and collisions have no visible side-effects. How do we know that in between two observed non-collision states objects really have touched each other? The answer is that collisions are detected on the basis of the observed motion behaviour. If two objects are approaching each other and are predicted to "collide" at a next instant, we assume in fact that a collision has happened when the following observed state matches the expected outcome.

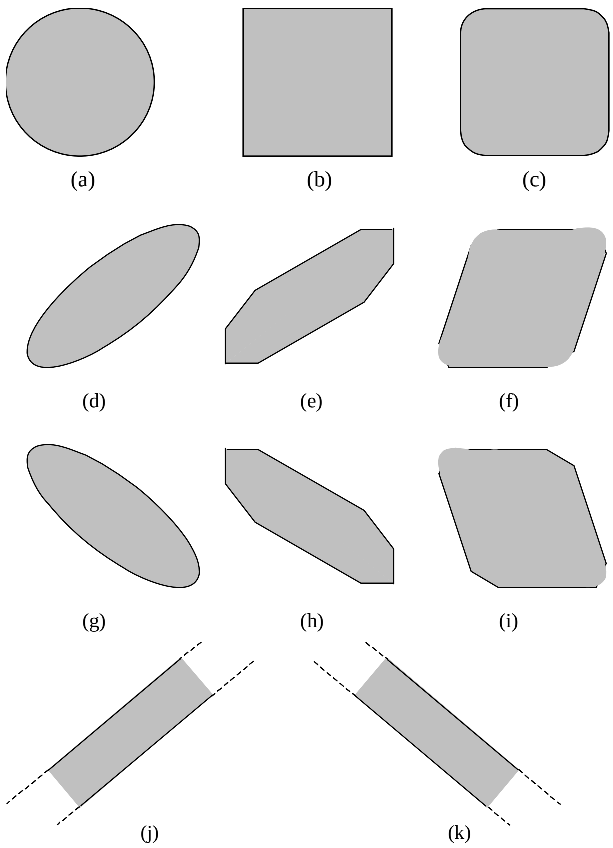

Figure 3. Typical forms of collision areas for (a) two circular objects, (b) two rectangular objects and (c) mixed rectangular/circular objects on a rectangular crossing $\gamma = 90°$; idem on a sharp crossing $\gamma < 90°$: (d-f) and on a wide crossing $\gamma > 90°$: (g-i). Collision areas (j-k) represent parallel cases $\gamma = 0°, 180°$.

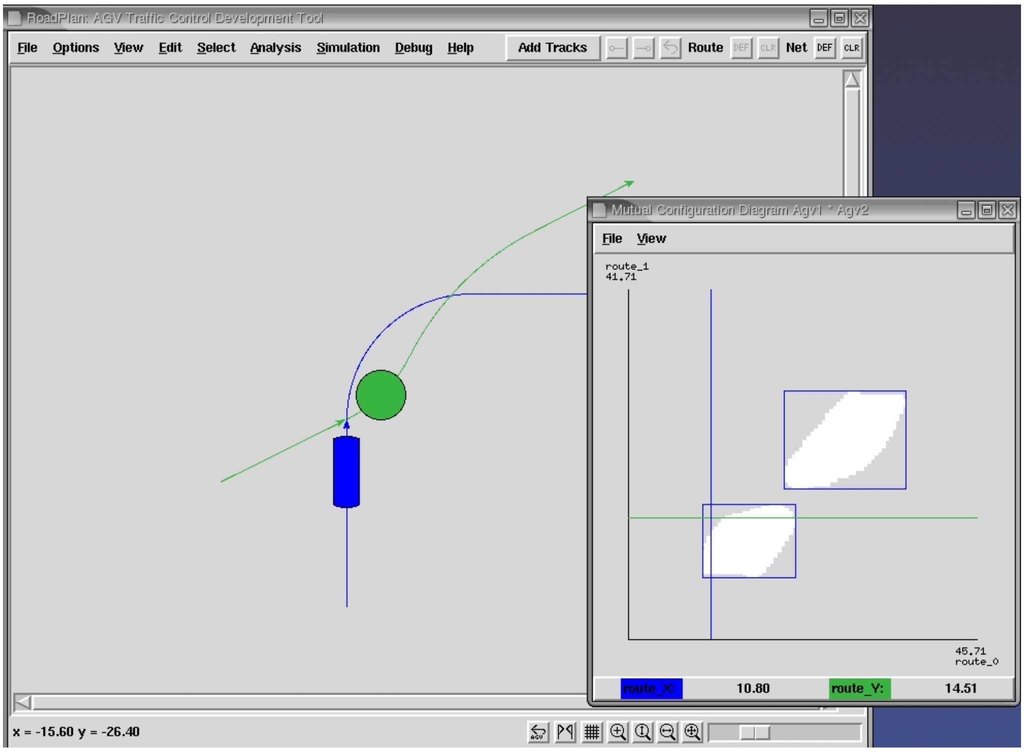

*Figure 4. Screenshot of the RoadPlan analysis tool showing the road map layout and the mutual configuration state diagram.
The white areas represent collision states. The current state of both "agv's" is indicated in the diagram by the orthogonal coordinate lines.*

As a consequence, collision detection depends on (1) an appropriate state prediction mechanism and (2) a collision correction model that computes the post-collision state adequately. The state prediction mechanism foresees the collision; the correction model confirms the collision.

3.1. Collision state prediction

A collision will occur if objects are in each others vicinity and at least one of the objects is approaching the other. If no (strong) forces are exerted on the objects, a collision prognosis can be based on the current positions and velocity vectors of the objects (at least on the short term) The velocity vectors define the intersection angle of the paths the objects follow[2]., whereas the current positions define the distances of the objects to the paths intersection point, or in case the paths are parallel, their relative translation distance.

Assuming simple 2D object shapes (such as a round ball and squares players), and given the intersection angle, the collision area is easily calculated using the analytically description that applies (see Figure 3). The distances to the centre of the intersection and the actual velocities now define the joint collision state and time instant completely as illustrated in Figure 5. Note that as we consider velocities to be constant (by neglecting the influence of forces on the short term); the joint motion trajectory in configuration space is a straight line. The slope depends on the relative speed of both objects: if object V_1 moves faster than V_2 the motion trajectory runs more "horizontally", in the reversed case more "vertically".

Of course, the most accurate prediction of the collision state is provided by the latest observation of a precollision state. This is the case when the state at the next observation instant is expected inside or even beyond the collision area.

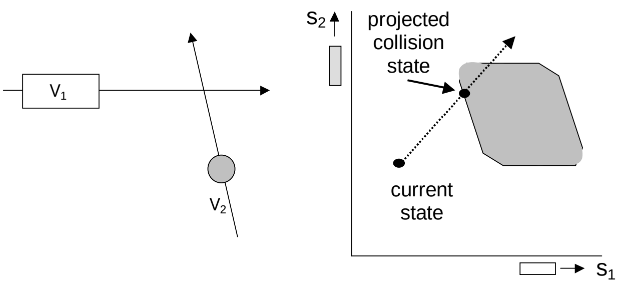

Figure 5. Prediction of the forthcoming collision instant based on the intersection of the joint motion trajectory with the collision area boundary. The slope of the straight motion trajectory is defined by the relative velocity of the objects.

[2] If one of the objects does not move, the path direction should be chosen according to its orientation.

3.2. Collision state correction

If a collision is assumed to have happened, one might just forget the past and recapture the motion state of the objects anew from the measurements. This has the drawback that motion state values are uncertain for some time. A better way is to model the effect of the assumed collision and use a collision state correction model to calculate the expected postcollision state. This has a number of advantages: (1) the actual occurrence of a collision can be checked and false collision detections can be ruled out, (2) the estimation of the motion state after the collision can be more accurate, in particular because (3) "collision physics" can be taken into account. The use of physical correction models is common in game programming. Application of a physical correction model in the MI20 robotsoccer system is brought up later in subsection 5.1.

4. COLLISION AVOIDANCE

4.1. As part of reactive control

In many cases collisions do not contribute in winning a game or in playing a fair match. For example, collisions between players of the same team clearly are not productive in any sense. Although collisions with opponent players can be useful in situations where blocking is needed, bluntly driving against opponents that are in the way is normally not an effective and desirable strategy. So, collision avoidance is an important part of the motion control.

Basically, the following options for collision avoidance exist: (1) velocity adaptation, (2) path deviation of one object only, (3) path deviation by both objects and (4) combined velocity and path adjustment. Option (3) could be useful for cooperating players of the same team in order to pass each other efficiently. Optimal passing strategies are considered in [6].

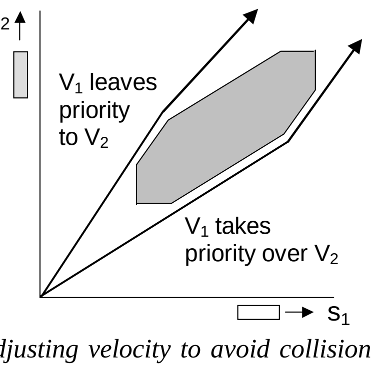

Figure 6. Adjusting velocity to avoid collision states. The velocity of V_1 is increased relative to the velocity of V_2 to take priority, or is decreased to give priority.

Adjusting velocity to avoid a collision is a common and practical solution in many circumstances. The configur-ation state approach can help in deciding what to do, either to accelerate and take priority or to decelerate and give priority. In the latter case even by making a full stop if needed. By identifying the collision area the relative velocity adjustment can be calculated just in order to avoid the collision states as can be seen in Figure 6. The velocity profile can be calculated with more or less subtle precision. For example, a “bumper sticking” profile can be designed to avoid collision by giving priority but with as little delay as possible.

4.2. As part of the path planning strategy

In the process of path planning, for instance to reach a target state in order to shoot or block a ball, care should be taking with respect to possibly interfering “obstacles”. In general, multiple solutions are possible and a search is made to find the best. Collision forecasting can help to eliminate “collision prone” plans. Especially if a planner is responsible to plan the motion of multiple player objects (say all “own” team members) mutual coordination of the motion plans is desired in order to avoid collisions. Coordination with non-controllable objects such as the opponent players makes less sense as their future motion behaviour is uncertain and can be guessed only in a limited way

Collision forecasting is based on comparing simultaneous motion plans. In contrast to short term collision prediction we can not restrict our motion model to straight paths and constant velocity anymore. A realistic motion plan consists of an elaborate path description and velocity profile. Typically a path is build out of a sequence of path sections, being either straight or curved line segments. The velocity profile could be specified for example by an initial velocity and, for each section, an acceleration constant.

If one compares two motion plans involving many paths sections, one would expect that collision analysis becomes a cumbersome task. However, collision analysis may be reduced to those parts of the mutual configuration space that “cover” the joint motion trajectory. The mutual configuration space is subdivided into “cross section areas”, one for each combination of sections of both paths. The velocity profiles determine which sections are visited simultaneously and which “zones” of the configuration space have to be analyzed, as shown in Figure 7. If we are interested in the first occurrence of a collision state the computation order may be chosen such that it runs along the trajectory.

The RoadPlan tool demonstrates that even a full mutual collision analysis of multiple paths can be performed quite efficiently. Generally in a majority of cases cross section areas are collision free and a simple test is enough to verify it.

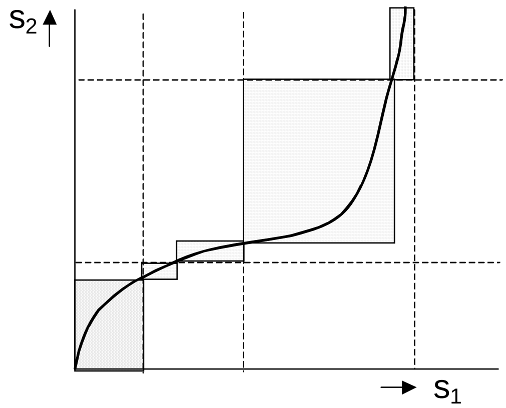

Figure 7. Only the zones that are passed by the joint trajectory need to be analyzed for collision states.

5. APPLICATIONS TO THE MI20 ROBOTSOCCER SYSTEM

5.1. Collision correction by the state estimator

The state estimator module of the MI20 robotsoccer control system has been enhanced by means of collision correction [3]. One of the goals of state estimation is to keep track of the team robots. As the robots have identical color patches, they can not be distinguished from each other by only taking a look at one image. Instead, the state estimator assigns an identifier to every team robot first and then tracks the robots by comparing successive images. Without a collision correction model, the state estimator often looses track when robots collide by mixing up the proper robot association. Robot identities are swapped which seriously degrades the soccer playing performance due to the wrong robot steering.

Collision correction is implemented in two stages. First, the actual precollision configuration state is reconstructed as described before. Second, a physical response model is applied by which the positions and velocities after the collision are calculated. The most important variable in a collision is the impulse factor, which depends on the velocities of the colliding objects, the collision normal, the masses of the objects and an elasticity factor e. This factor ranges from e = 1, a perfectly elastic collision where the objects bounce immediately, to e = 0, a perfectly inelastic collision where the objects will stick together.

The evaluation of the correction for collisions between robots showed that almost no identifier swapping occurred anymore. It is reported in [3] that without any correction a specific collision test scenario lead to erroneous identifier swapping in 24 of the 50 runs. With precollision correction only 3 runs gave an error, while with additional physical correction the test showed no errors. So, the pre-collision correction accounts for the biggest difference. The influence of the physical model is small but certainly helps.

It may be the case (see for instance the example in Figure 8) that a collision may govern the motion behaviour over quite a long period of observations. Measurements show that with physical correction during the collision phase the state estimator produces significantly smaller positional errors than without correction.

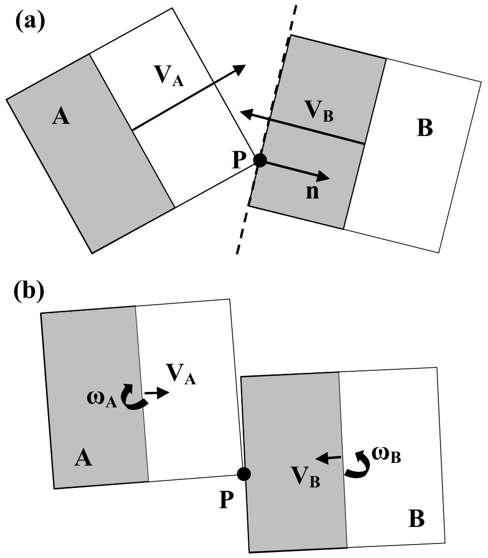

Figure 8. Example of collision response situation: (a) initial state, (b) resulting state.

5.2. Collision prediction between planned motion trajectories

In the MI20 system paths have been planned with S-curves [7]. These paths are described piecewise by sections of constant curvature. A method has been implemented to forecast collisions for predicted motion trajectories of two vehicles [4]. The collision analysis is based on the core routines of the RoadPlan program tool. An extension is made to specify velocity profiles and associate them with paths. Based on the geometry of the paths velocity profiles are predicted as realistically as possible.

The existing path planning method evaluates four S-curves that connect a start pose and a target pose. The S-curve that has the shortest length was selected. A new planning strategy is made that uses the forecast module to test for collisions on the S-curves prior to selecting them. At first the shortest curve is tested. If the curve is collision free it is selected, else the process is repeated with the next longer alternative. If all curves are predicted to cause a collision, the shortest curve is used.

By using a forecast on the predicted trajectories more information on the efficiency of them can be obtained. For example, the motion trajectory can provide the time of completion of the path assuming some delay in avoiding a collision.

As an example, in Figure 9 the screen output is shown of a collision forecast analysis of two “S-curve” plans performed by RoadPlan.

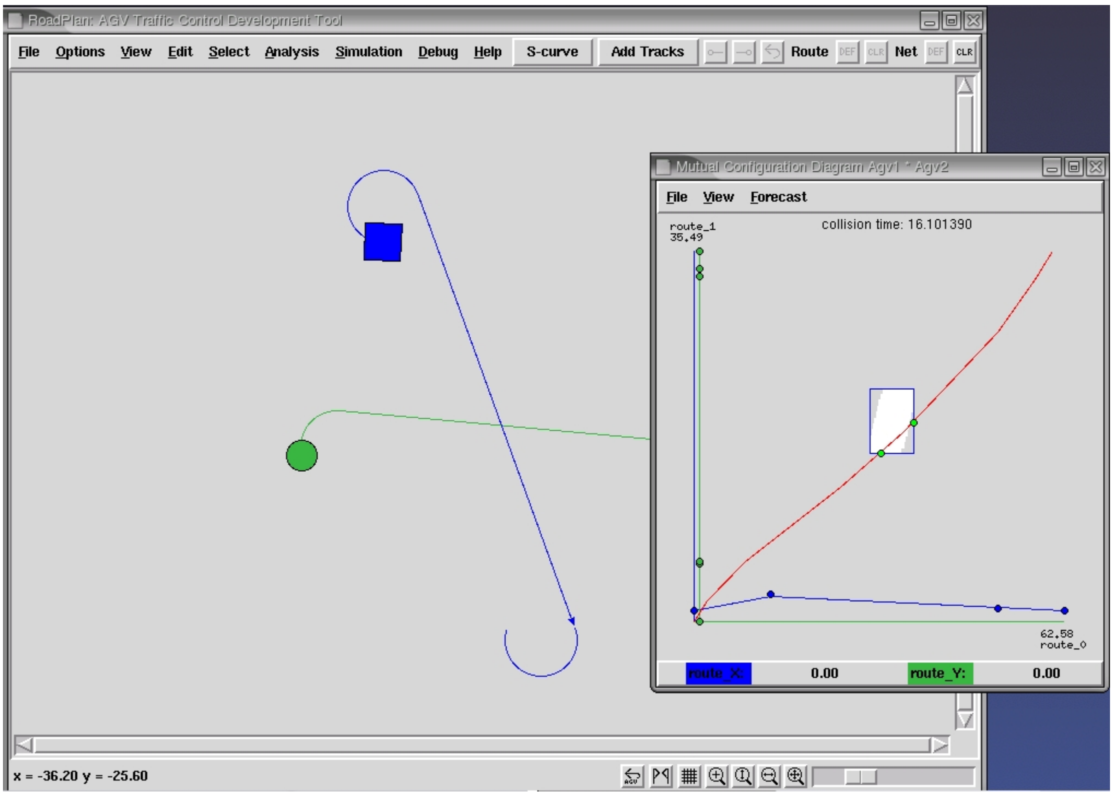

Figure 9. Comparison of vehicles driving along S-curves. The joint motion trajectory shown in the mutual configuration diagram resulted from the (interactively manipulated) velocity profiles. Note that a collision is forecasted at time 16,101….

6. CONCLUSION

The mutual configuration state approach offers a versatile method to predict the occurrence of collisions between simple shaped moving objects. It is shown how collisions can be detected and how the state estimation can be improved by taking corrective actions. It is indicated how reactive control processes can take anticipatory measures such that collisions are avoided. Also collision forecasting is suggested as helpful tool for planning processes in their selection of collision free motion trajectories.

Application of collision prediction analysis in the MI20 robotsoccer control system shows that improvements can be reached in some respects. These applications should be considered as preliminary investigations of the usefulness of collision analysis based on the configuration approach. The accuracy of the robotsoccer control system has to be improved first before we can judge properly to what extent the gains of "collision monitoring" enhance the playing skills and make the additional overhead worthwhile. Smart collision avoidance methods will become anyway a "must" if playing rules are made more restrictive and say attacking opponent players is penalized. It would stimulate research on this interesting subject.

7. REFERENCES

[1] G. Dudek, M. Jenkin, "Computational principles of mobile robotics, Cambridge University Press, 2000.

[2] MI20 Robot soccer team, "Mobile Intelligence Twenty", look at http://www.robosoccer.nl, [in preparation]

[3] S. Grond, "State estimation of colliding objects in a robotsoccer environment", Master's Thesis, Department of Computer Science, University of Twente, 2005.

[4] J. van der Linden, "Dynamic avoidance control of soccer playing mini-robots", Master's Thesis, Department of Computer Science, University of Twente, 2005.

[5] A.L. Schoute, "RoadPlan manual version 3.1.", Technical report TR-CTIT-02-28, Centre for Telematics and Information Technology, Univ.of Twente, The Netherlands, 2002.

[6] A.L. Schoute, "Time-optimal collision avoidance of automatically guided vehicles", Eurocast 2005, 10th International Workshop on Computer Aided Systems Theory, 2005.

[7] T. Verschoor, "Dynamic motion planning of soccer playing mini-robots", Master's Thesis, Department of Computer Science, University of Twente, 2004.