The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Generating collision free reaching movements for redundant manipulators using dynamical systems

Hendrik Reimann, Ioannis Iossifidis and Gregor Schöner

Abstract— **For autonomous robots to manipulate objects in unknown environments, they must be able to move their arms without colliding with nearby objects, other agents or humans. The simultaneous avoidance of multiple obstacles in real time by all link segments of a manipulator is still a hard task both in practice and in theory. We present a systematic scheme for the generation of collision free movements for redundant manipulators in scenes with arbitrarily many obstacles. Based on the dynamical systems approach to robotics, constraints are formulated as contributions to a dynamical system that erect attractors for targets and repellors for obstacles. These contributions are formulated in terms of variables relevant to each constraint and then transformed into vector fields over the manipulator joint velocity vector as an embedding space in which all constraints are simultaneously observed. We demonstrate the feasibility of the approach by implementing it on a real anthropomorphic 8-degrees-of-freedom redundant manipulator. In addition, performance is characterized by detecting failures in a systematic simulation experiment in randomized scenes with varying numbers of obstacles.**

I. INTRODUCTION

Autonomous robotic agents acting in environments shared with other agents and humans must be capable of generating movements that reach a desired target while avoiding collision with objects and other entities. The scenes an autonomous robot acts in are usually previously unknown, can be cluttered and subject to sudden changes. These conditions require schemes of behavior generation that depend only on locally available data and produce appropriate actions sufficiently fast to adapt to changing situations.

One theoretical difficulty in the generation of autonomous behavior is that the number of constraints quickly surpasses the available degrees of freedom [1]. As an additional complication, different constraints often act on different state variables of the system. Solutions to this problem are usually tailor-made for specific scenarios and do not apply to other tasks. To our knowledge, there is no general solution that applies across a wide range of constrained tasks for autonomous agents.

The dynamical systems approach to behavior generation deals with these problems by formulating behavioral constraints as attractors or repellors in a vector field over a chosen behavioral variable of a robotic system. The flow of the vector field is used to generate trajectories for the behavioral variables, which are realized by a low-level motor control module. The dynamical systems approach was successfully applied to steer vehicles in a plane cluttered with obstacles [2] and generate reaching movements for a particular 8 DoF redundant manipulator [3]. It was also used in a psychological model describing human walking paths in changing environments [4].

In these scenarios, the dynamical systems approach is able to deal with the problem of multiple different constraints by formulating each constraint as an attractor or repellor for a small set of predefined behavioral variables. This allows combining the constraints by superposition of the corresponding vector fields, producing behavior that adheres to all constraints simultaneously.

Here we attempt a systematic solution to the problem of generating reaching movements to a target for arbitrary redundant manipulators while avoiding collision between all link segments of the manipulator and any obstacle, using the dynamical systems approach. To reach the target, the movement direction of the end-effector is important. For avoiding collisions between a link segment and an obstacle, the relevant variables are position and direction and magnitude of the link segment's movement relative to the obstacle. Because these different constraints apply to different segments of the manipulator, and the application points vary over time, it is not feasible to find a small number of behavioral variables that are relevant to all constraints throughout the movement.

To solve this problem, we use two layers of state variables. Each constraint is formulated as an attractor or repellor in a behavioral variable that is relevant locally. This term is then transformed into a vector field over the joint velocities as a common variable for all constraints. Over this common state variable, it is possible to superpose the vector fields in order to generate behavior that adheres to all contributing constraints.

Related work

There is sizeable literature on the planning of arm trajectories (summarized by [5] and [6]), which includes sophisticated exact approaches (e.g. [7]). These typically require detailed metric information about the scene and are static in nature, requiring replanning when the scene changes [8]. The potential field method, originally developed by Khatib [9] (see also [10]), is probably the heuristic approach most adapted to dealing with dynamic environments as well as with imprecise information (see the more recent work of Khatib's group, e.g. [11], [12]). Another closely related solution was proposed by Maciejewski and Klein [13], who maximize the distance between the whole manipulator and an obstacle by computing a velocity vector that moves the

The authors are with the Institut für Neuroinformatik, Ruhr-Universität Bochum, 44801 Bochum, Germany {reimann | iossifidis | schoner}@neuroinformatik.rub.de

'avoidance point', the point on the manipulator closest to the obstacle, away from the obstacle. To integrate this vector with the predetermined end-point trajectory, the constraint equations are combined and a suitable pseudo-inverse is applied. In comparison, the attractor dynamics approach presented here enables us to include multiple obstacles and use one 'avoidance point' for each link segment rather than only one for the whole manipulator.

1) Artificial Potential Field Approach: The Artificial Potential Field Approach (APFA) was introduced by Khatib for manipulator movement generation [9], though it has seen widest use in vehicle path planning. Artificial potential fields that generate attractive forces towards the target and repelling forces away from obstacles are erected in the scene. Acting like physical forces on each link segment, they are transformed into joint torques and then joint accelerations to generate a movement trajectory.

Our approach is formally similar to the APFA in that both schemes make use of a vector field over velocity space to generate movement trajectories. The conceptual difference is that in the APFA this vector field only depends on the position of objects in the scene, while the dynamic systems approach treats the velocity vector as a crucial second variable affecting the shape of the vector field. While due to the conceptual inheritance from real potential functions the APFA has rather strict requirements on the vector fields it generates, the dynamic systems approach allows more flexibility in the choice of relevant variables and design of attractor landscapes. Critically, using the joint velocity vector as a dynamical variable makes it possible to generate movement while the system is in or close to an attractor. For instance, a constant value for the absolute magnitude of the end-effector velocity, which can easily be achieved by a fixed-point attractor, is sufficient to keep the arm moving. Similarly, the heading direction of the end-effector in workspace is related to the joint velocity vector and can have a constant or slowly varying value while the arm is moving. That value can be generated from a fixed point attractor for heading direction.

As a result, designing a dynamics of the joint velocity vector that fulfills the constraints of obstacle avoidance and target acquisition only requires that the system stays in an appropriate attractor from an initial configuration. There is no need to ensure that any transient toward an attractor fulfills the constraints. This is in contrast to the APFA, in which only the target state is an attractor. The movement is generated as a transient from an initial state to that terminal attractor. In this case, spurious attractors emerge near obstacles, in which the system may get caught, a well known problem, which can be overcome only with considerable effort (see e.g. [14], [15]).

A practical advantage over the APFA is that the scheme presented here is purely kinematic, the technically demanding step of transforming the potential field forces to torques and then to accelerations, requiring values for the inertial properties of the manipulator, are not necessary.

2) Dynamic Movement Primitives: Another movement generation scheme making strong use of dynamical systems are the Dynamic Movement Primitives (DMP) by Ijspeert et al. [16]. In this framework, trajectories are learned from demonstration by a teacher and can then be generalized to new targets. In a recent paper, Park et al. [17] adapted DMP to include obstacle avoidance using the APFA for end-effector trajectory generation and an inverse velocity kinematics model that constrains the null space to collision free configurations to avoid link segment collision [13]. Park et al. also realized the advantage of including velocity information to shape the vector field and modified their potential functions accordingly. Hoffmann et al. [18] combine the DMP with angular obstacle avoidance based on the dynamic systems approach on the end-effector level, although without including link segment avoidance.

A key difference is that in the DMP framework a movement towards a target is planned ahead of movement onset as an explicit function of time, and is then combined with other influences, e.g. from obstacles. This is a part of the strategy of determining movement primitives from imitation, in which movement trajectories learned from demonstrated samples are transformed to new initial conditions and new targets. To the best of our knowledge, generalizing this imitation strategy to combinations of multiple constraints has not been met with success yet (see, however, [19] for a first attempt to do obstacle avoidance in this framework).

II. BEHAVIOR GENERATION

The desired behavior for the manipulator is to acquire the given target while avoiding collision with obstacles. We formulate each behavioral constraint as an attractor or repellor in some locally relevant dynamical variable.

In the following paragraphs, italic letters indicate scalars, bold letters indicate vectors or points in three-dimensional cartesian space, bold capital letters are elements of n-dimensional joint space, and other entities are denoted by italic capital letters.

As a sigmoid function that rises smoothly from 0 to 1 between two thresholds a, b, we use

$$\sigma_{a,b}(x) = \begin{cases} 0 & : \quad x \leq a, \\ -\frac{1}{2}\cos\left(\frac{x-a}{b-a}\pi\right) + \frac{1}{2} & : \quad a < x < b, \\ 1 & : \quad b \leq x. \end{cases} \qquad (1)$$

The letter α_* always indicates a gain factor relevant to $*$.

A. Target acquisition

Similar to a vehicle steering towards a goal, the target acquisition is pursued by changing the heading direction of the end-effector towards the direction of the target. The target angle $\phi = \arccos(\frac{\langle \mathbf{v},\mathbf{k}\rangle}{|\mathbf{v}||\mathbf{k}|})$ is defined as the angle between the end-effector velocity vector $\mathbf{v}$ and the direction $\mathbf{k} = \mathbf{g} - \mathbf{p}$ from the end-effector position $\mathbf{p}$ towards the target $\mathbf{g}$. We define a vector field over ϕ

$$f_{dir} = -\alpha_\phi \sin\phi \qquad (2)$$

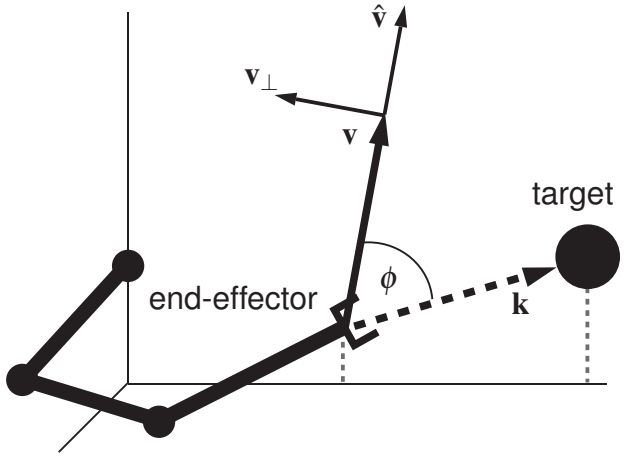

Fig. 1. Schematic of how the target angle ϕ, target direction $\mathbf{k}$ and directions of the cartesian acceleration vectors $\mathbf{v}_\perp$ and $\hat{\mathbf{v}}$ contributing to the vector field for target acquisition are defined.

that has a single attractor at $\phi = 0$, meaning the flow of the vector field makes the end-effector move towards the target.

Changing the target angle ϕ corresponds to accelerating $\mathbf{v}$ in direction of

$$\mathbf{v}_\perp = \left(\mathbf{k} - \frac{\langle \mathbf{k}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v}\right) \frac{|\,\mathbf{v}\,|}{|\,\mathbf{k} - \frac{\langle \mathbf{k}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v}\,|}, \tag{3}$$

which is the vector orthogonal to $\mathbf{v}$ in the plane spanned by $\mathbf{v}$ and $\mathbf{k}$ (see fig. 1), so the vector field

$$\mathbf{f}_{dir} = \mathbf{v}_\perp \cdot f_{dir} \tag{4}$$

for the cartesian end-effector velocity realizes the desired change of target angle f_{dir}. Using the Moore-Penrose pseudo inverse J^+ of the end-effector Jacobian J yields a vector field in joint space

$$\mathbf{F}_{dir} = J^+ \cdot \mathbf{f}_{dir} \tag{5}$$

that changes the heading direction of the end-effector towards the target.

In a similar manner, we define a vector field for the absolute path velocity $v = |\,\mathbf{v}\,|$ of the end-effector as

$$f_{vel} = -\alpha_{vel}(v - v_{des}), \tag{6}$$

with a single attractor at a desired value v_{des} for end-effector velocity. A corresponding vector field for the cartesian velocity vector is

$$\mathbf{f}_{vel} = \hat{\mathbf{v}} \cdot f_{vel}, \tag{7}$$

with $\hat{\mathbf{v}} = \frac{\mathbf{v}}{|\,\mathbf{v}\,|}$, and again the vector field in joint space is given by

$$\mathbf{F}_{vel} = J^+ \cdot \mathbf{f}_{vel}. \tag{8}$$

In close vicinity of the target, heading direction and path velocity are not suitable coordinates for behavior generation anymore, so for small distances $d = |\,\mathbf{g} - \mathbf{p}\,|$, we switch to an attractor in cartesian coordinates, given by the damped harmonic oscillator

$$\mathbf{f}_{pos} = -\alpha_v\big(\mathbf{v} - \alpha_p(\mathbf{g} - \mathbf{p})\big), \tag{9}$$

and again, this is lifted to joint space by

$$\mathbf{F}_{pos} = J^+ \cdot \mathbf{f}_{pos}. \tag{10}$$

A well known problem in second order approaches to redundant manipulator control is that velocities in the task variable null space may build up over longer movements [20]. As persistent null-space velocities are undesired after the target has been reached, we introduce a homogeneous damping term

$$\mathbf{F}_{damp} = -\alpha_{damp}\dot{\theta}. \tag{11}$$

This weak damping reduces any velocities previously built up in the task null space. As a drawback, this term counteracts control in task space, effectively acting as a perturbation. The stabilization of the task variable (eq. 9) limits the effect of his perturbation.

To finalize the target acquisition, we add up these vector fields to

$$\mathbf{F}_{tar} = (1 - \sigma_d)\Big(\mathbf{F}_{dir} + \mathbf{F}_{vel}\Big) + \sigma_d\Big(\mathbf{F}_{pos} + \mathbf{F}_{damp}\Big), \tag{12}$$

with $\sigma_d = \sigma_{d_1,d_2}(d)$ shifting the weight from directional control towards the positional fine control near the end of the movement.

B. Obstacle avoidance

In order to prevent collision of manipulator link segments with obstacles in the scene, we change the movement vectors of link segments away from directions in which obstacles are positioned. To describe the vector field that achieves this, we first define the direction in which the repelling force acts, and then its magnitude depending on the distance and current movement states.

Both link segments and obstacles are enclosed in cylindrical bounding volumes, topped off with half-spheres at the ends. Obstacle cylinders are always oriented towards the z-axis. For any link segment S and obstacle O, let $\mathbf{s}$ and $\mathbf{o}$ be the points on their respective bounding volumes with minimal distance to each other.

1) Avoidance direction: To find a candidate for the direction of the repelling force, we look for a vector that is perpendicular to the movement vector of the segment point $\dot{\mathbf{s}} = \mathbf{v}_s$ and points away from the obstacle. To define this mathematically, let N be the plane that is normal to $\mathbf{v}_s$. An orthonormal base $\mathbf{u}_1, \mathbf{u}_2$ of N is given by $\mathbf{u}_2' = \mathbf{e}_3 - \frac{\langle \mathbf{v}_s, \mathbf{e}_3 \rangle}{\langle \mathbf{v}_s, \mathbf{v}_s \rangle} \mathbf{v}_s$, $\mathbf{u}_1' = \mathbf{v}_s \times \mathbf{u}_2$, $\mathbf{u}_i = \frac{\mathbf{u}_i'}{|\,\mathbf{u}_i'\,|}$, in which the second base vector is the projection of the z-Axis vector $\mathbf{e}_3$ onto N.

Let $\mathbf{a}$ and $\mathbf{b}$ be the projections of the center points of the half-spheres at the obstacle bounding volume top and bottom to N. Due to obstacles always standing upright and the choice of the base $\mathbf{u}_1, \mathbf{u}_2$, the projected points $\mathbf{a}$ and $\mathbf{b}$ lie on a vertical line in N. Assume without loss of generality that $\mathbf{a}$ is the upper point, i.e. $\langle \mathbf{a}, \mathbf{u}_2 \rangle > \langle \mathbf{b}, \mathbf{u}_2 \rangle$, then the point on that line segment with minimal distance to the origin is given by

$$\mathbf{q} = \begin{cases} \mathbf{a} & : \ \langle \mathbf{a}, \mathbf{u}_2 \rangle < 0, \\ \mathbf{b} & : \ \langle \mathbf{b}, \mathbf{u}_2 \rangle > 0, \\ \langle \mathbf{b}, \mathbf{u}_1 \rangle \mathbf{u}_1 & : \ \text{otherwise}, \end{cases} \tag{13}$$

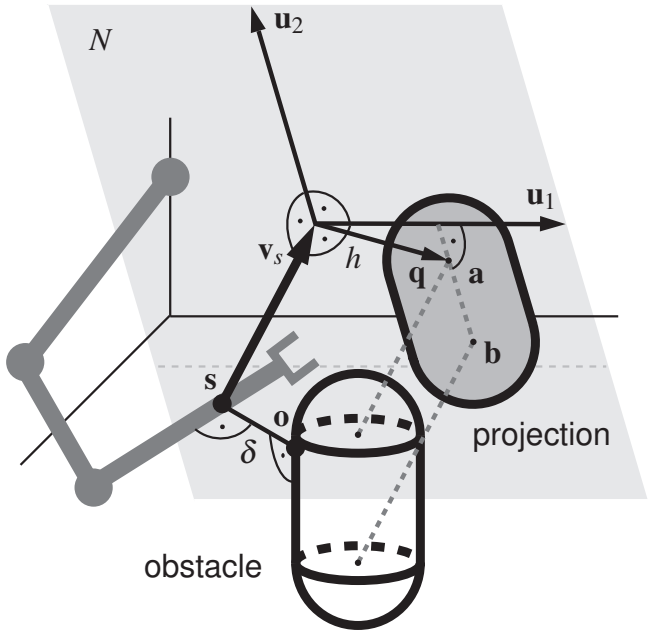

Fig. 2. Projection of an obstacle to the normal plane N of the velocity $\mathbf{v}_s$ of the segment point $\mathbf{s}$ with minimal distance δ to a point $\mathbf{o}$ on the obstacle. N is spanned by the orthonormal base $\mathbf{u}_1, \mathbf{u}_2$, the projection of the obstacle center line segment is the line segment between $\mathbf{a}$ and $\mathbf{b}$ in N. The smallest distance of any point on this line segment to the projection of $\mathbf{v}_s$ is given by $\mathbf{q}$, with h indicating the distance to the obstacle projection boundary.

and the distance of the obstacle bounding volume projection to the origin is

$$h = |\,\mathbf{q}\,| - r, \tag{14}$$

where r is the radius of the obstacle bounding volume. Note that 'origin' here means the origin of N, which is the projection of $\mathbf{v}_s$, and all coordinates are assumed to be in N. Fig. 2 shows the plane N and how an obstacle is projected.

The 'best' avoidance direction, i.e. the one changing the movement direction directly away from the obstacle direction, is given by $-\mathbf{q}$. For some situations, though, this is not a suitable choice: when the link segment is moving horizontally in a plane that intersects the cylinder part of the obstacle bounding volume, $\mathbf{q}$ also lies in that plane, resulting in a change of link segment movement to one side. If the link segment is near the base of the kinematic chain, this is unlikely to result in a path in which the whole manipulator avoids the obstacle, because either the immobile base or the rest of the arm up to the end-effector are still on one side of the obstacle, while the link segment in question is trying to avoid it by moving around the other side.

To prevent this kind of deadlock, instead of choosing $\mathbf{q}$ itself as avoidance direction, we use a vector that also lies in N, but is rotated towards $\mathbf{u}_2$ by an amount depending upon the location of the link segment along the kinematic chain. The angle of $-\mathbf{q}$ with $\mathbf{u}_2$ is given by

$$\gamma = \arccos\left(\frac{\langle -\mathbf{q}, \mathbf{u}_2\rangle}{|\,\mathbf{q}\,|}\right). \tag{15}$$

For the j-th link segment of kinematic chain with n joints,

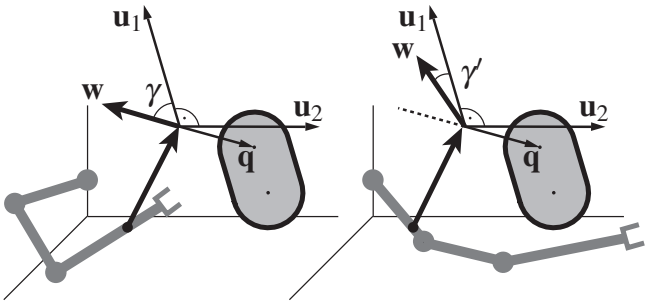

Fig. 3. The avoidance direction depends upon the link index. Links that are far away from the base can avoid an obstacle in the direction $\mathbf{w} = -\mathbf{q}$ that points directly away from the obstacle (left). For link segments close to the base, the avoidance direction is tilted upwards (right).

we decrease that angle by applying the ad-hoc function

$$\gamma' = \frac{3}{4}\frac{j-2}{n-2}\gamma, \tag{16}$$

and set

$$\mathbf{w} = \begin{cases} -\sin\gamma'\,\mathbf{u}_1 + \cos\gamma'\,\mathbf{u}_2 & : \quad \langle \mathbf{q}, \mathbf{u}_1\rangle \geq 0, \\ +\sin\gamma'\,\mathbf{u}_1 + \cos\gamma'\,\mathbf{u}_2 & : \quad \langle \mathbf{q}, \mathbf{u}_1\rangle < 0, \end{cases} \tag{17}$$

which is $-\mathbf{q}$ rotated towards $\mathbf{u}_2$ so that $\angle(\mathbf{w}, \mathbf{u}_2) = \gamma'$. Fig. 3 illustrates this dependency of avoidance direction upon the link segment index in the chain.

Using this $\mathbf{w}$ as avoidance direction is suitable for most situations. Only for the first segment it does not make sense, as the direction this segment moves to cannot be chosen with sufficient freedom. We thus set

$$\mathbf{w} = -\frac{\mathbf{v}_s}{|\,\mathbf{v}_s\,|} : j = 1, \tag{18}$$

essentially just braking the first segment when it approaches an obstacle.

2) Magnitude of repelling force: Defining the strength of the avoidance action for a given situation boils down to deciding how likely an obstacle collision is in that situation. Three factors play a role, the distance between the link segment and the obstacle, the movement direction of the link segment and how fast it is moving in that direction. For each of these factors, we define a weight factor, and the product of these weight factors will give the magnitude of the repelling force.

For the dependency on distance $\delta = |\,\mathbf{o} - \mathbf{s}\,|$, define

$$w_\delta = \big(1 - \sigma_{\delta_1,\delta_2}(\delta)\big)\frac{\delta_1}{\delta}, \tag{19}$$

which is zero for distances $\delta > \delta_2$, 1 for $\delta = \delta_1$ and grows towards $+\infty$ for $\delta \to 0$.

For the movement direction dependency, let $\mathbf{m}$ be the point on the line $\mathbf{s} + \lambda\,\mathbf{v}_s, \lambda \in \mathbb{R}$, that is closest to the obstacle, and $\mathbf{o}_m$ the point on the obstacle bounding volume with minimal distance to $\mathbf{m}$. Define the obstacle angle as

$$\psi = \operatorname{atan}_2(|\,\mathbf{m} - \mathbf{o}_m\,|, |\,\mathbf{m} - \mathbf{s}\,|), \tag{20}$$

which is the minimal angle between $\mathbf{v}_s$ and any vector going through or touching the obstacle bounding volume. Set

$$w_\psi = \begin{cases} 0 & : \quad \langle \mathbf{v}_s, \mathbf{o} - \mathbf{s} \rangle \leq 0, \\ 1 - \sigma_{\psi_1,\psi_2}(\psi) & : \quad \text{otherwise}, \end{cases} \tag{21}$$

which vanishes if $\mathbf{v}_s$ is zero or ψ is too large.

Finally, as a dependancy on the movement speed we just take that value itself as

$$w_v = |\mathbf{v}_s|. \tag{22}$$

With these three weight factors depending on distance, movement direction and speed, we can define the magnitude of the repelling force as of the obstacle O on the segment S as

$$f_O^S = \alpha_{obs} \cdot w_\delta \cdot w_\psi \cdot w_v. \tag{23}$$

3) Obstacle vector field: Having defined an avoidance direction in $\mathbb{R}^3$ and a magnitude f_O^S, what remains is to define a corresponding vector field in joint space. Let

$$J_\mathbf{s} = \left(\frac{\partial \mathbf{s}_i}{\partial \theta_j} \right)_{i,j} \tag{24}$$

be the Jacobian of the segment point $\mathbf{s}$. Then $J_\mathbf{s}^\mathbf{w} = \mathbf{w}^T J_\mathbf{s}$ gives the change along $\mathbf{w}$ by changes of θ. We use the pseudo inverse of this to define

$$\mathbf{F}_O^S = f_O^S \cdot (J_\mathbf{s}^\mathbf{w})^+, \tag{25}$$

which is a vector in joint space that realizes the desired change in direction $\mathbf{w}$ with magnitude f_O^S .

For the complete vector field of one obstacle, we just sum up the vector fields for each of the n link segments S_j, getting

$$\mathbf{F}_O = \sum_j \mathbf{F}_O^{S_j}. \tag{26}$$

C. Overall behavior

In a complex scene with a target and multiple obstacles O_i, we can now define a total vector field to generate behavior that adheres to all behavioral constraints, i.e. moving the end-effector to the target without hitting an obstacle with any part of the manipulator, by just summing up the vector fields for each of these behavioral constraint, setting

$$\mathbf{F} = \mathbf{F}_{tar} + \sum_i \mathbf{F}_{O_i}. \tag{27}$$

This is a vector field over the joint velocity space, behavior is generated by using the value of $\mathbf{F}$ for the current state as joint acceleration vector

$$\ddot{\theta} = \mathbf{F}. \tag{28}$$

The resultant joint acceleration vector is numerically integrated twice to generate a joint angle vector, which is then realized by the hardware servo controllers.

III. IMPLEMENTATION AND RESULTS

The behavior generation scheme described in the previous section was implemented for the 8 DoF manipulator CoRA [21]. Experiments that demonstrate successful generation of movements adhering to the given behavioral constraints were carried out both with the real manipulator and in a software simulator. The parameter values used in the experiments can be found in section V.

A. CoRA

The robotic assistant system CoRA has an anthropomorphic seven degrees of freedom arm mounted on a one degree of freedom trunk. CoRA is built from a modular robotics system, in which each module is servo-controlled and communicates via a CAN-bus interface with the controlling PC. Above the trunk a two DoF pan/tilt unit carrying a stereo color camera system and microphones is assembled.

The behavior generation scheme was capable of producing satisfying movement trajectories for CoRA for scenes with several obstacles (see fig. 4).

B. Randomized scenes

For a more systematic examination of the behavior generation scheme, it was tested in a sequence of randomly generated scenes in the software simulator of CoRA. In each trial, the initial configuration was randomized (normal distribution). The target was placed randomly in a predetermined area, and a varying number of obstacles was also placed randomly in an area ranging from the manipulator to around the target (uniform distributions). No obstacle was placed closer than 9cm to the target, allowing a minimal leeway of 3cm for the end-effector, enclosed in a 6cm radius bounding volume, to reach it. Radius and height of the obstacles was also randomized, while the manipulator starting configuration was fixed at θ_{init} (see fig. 5).

Table I shows the results of the experiment. A successful trajectory to the target was found in the majority of all trials, overwhelmingly so for small numbers of obstacles and decreasing only significantly as the scenes get cluttered. An investigation of the failed trials implied that the reasons for failing to find a path to the target fall in fairly distinct categories, which are described below, listed with the reference letter used in the table.

- P: The most prevalent reason was a single high obstacle near the base of the manipulator that prevented the link segments close to the base to move towards the target, reducing the effectively reachable workspace significantly. This resulted in failures when the target was located far out in the workspace, though for closer targets successful trajectories could still be found (an example of this situation is shown in fig. 5).
- W: In a small number of cases, the manipulator reached the end of the workspace while moving around an obstacle, and then failed to find a way back.
- O: For certain combinations the target was so obstructed by the obstacles that a configuration that reached the target without collision was nonexistent or very hard to find.

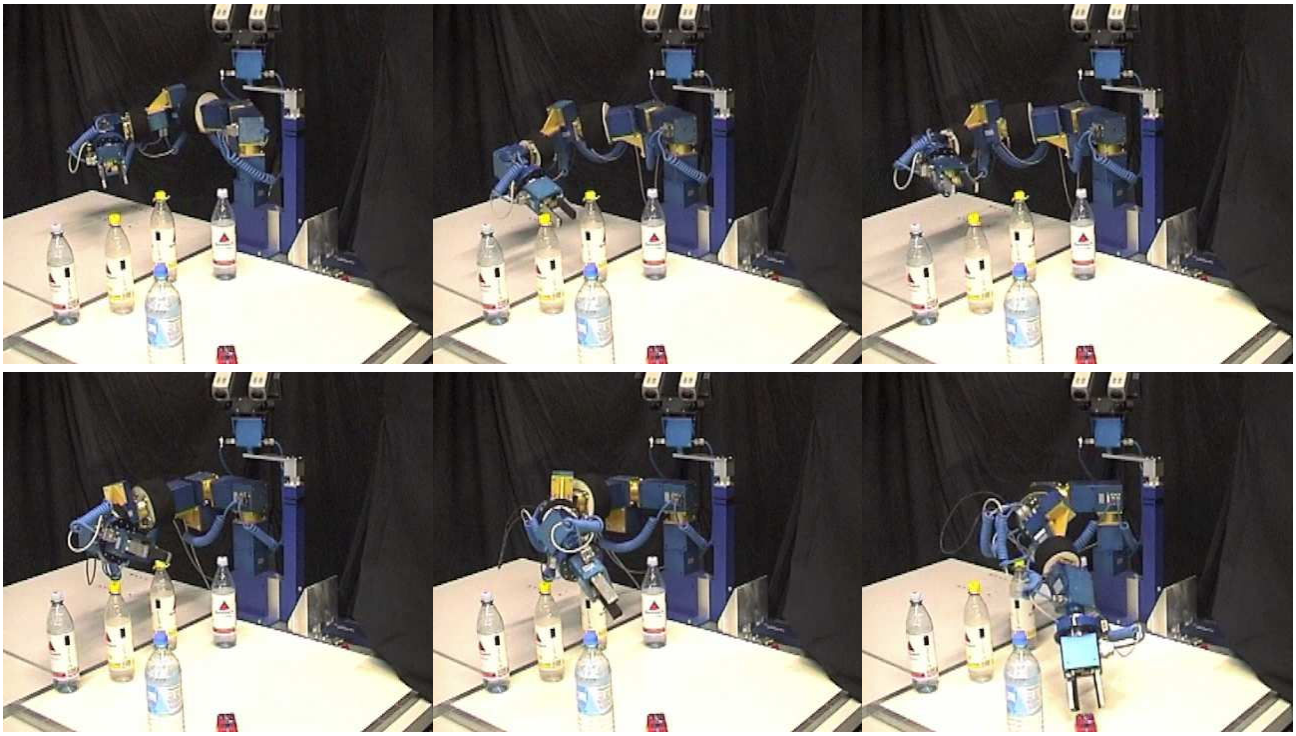

Fig. 4. Sequence of CoRA successfully reaching for a toy car while avoiding collision with five obstacles in the scene.

Failures of this kind did not occur for small numbers of obstacles, and only became frequent in very cluttered scenes.

X: When one link segment simultaneously approached two or more obstacles on different sides, the avoidance directions partly cancelled each other out, until the obstacle distance had become very small and the corresponding factor w_δ in the repelling vector field so large that the simulation became numerically unstable.

D: When the manipulator avoids an obstacle with a link segment closer to the base, this results in persisting motion in the null space of the end-effector even after the obstacle has been passed. In situations where only a tight path towards the target is available, that null space drift prevented the manipulator from successfully entering that path.

The first three categories of failures are instances of the general situation where a path that reaches the target without collision might exist, but can only be found by first realizing a significant change of manipulator configuration. As the behavior generation scheme presented here is a local approach, this shortcoming is to be expected to some degree. A possible way to prevent the numerical explosions would be to include a more sophisticated method of regulating the absolute velocity than the one given by eq. (6), that reduces the overall velocity in the vicinity of obstacles which allows more time to change the movement direction away from possible collision paths. This is not the focus of the present paper, though. The persistance of null-space drift after obstacles have been cleared is a real problem, though more a theoretical one as the small number of occurences indicates, and we plan to investigate different methods of solving it in the future.

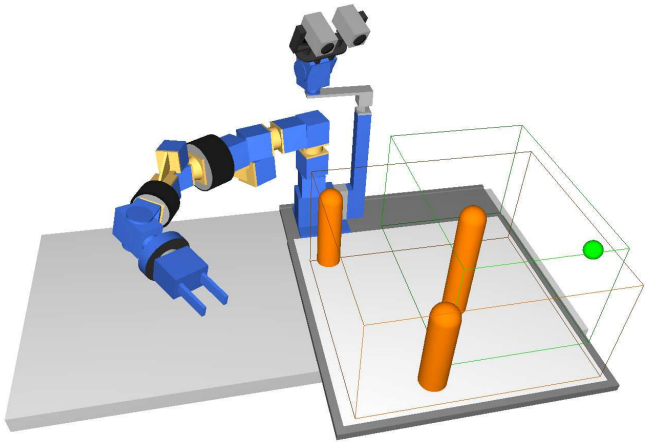

Fig. 5. Randomly generated scene with three obstacles. The manipulator is in the reference configuration θ_{init}. The target was randomly placed in the green box, the obstacles distributed over the orange box.

C. Special cases

In addition to the randomized scenes, we set up two special cases with obstacle configurations where it would seem particularly hard to find a successful trajectory as qualitative demonstration of the practical applicability of our presented scheme. In the first case, the target is encircled within several large obstacles, leaving only a narrow path from above for the manipulator to reach through (fig. 6, upper panel). The second scene is a border case of the failed random trials,

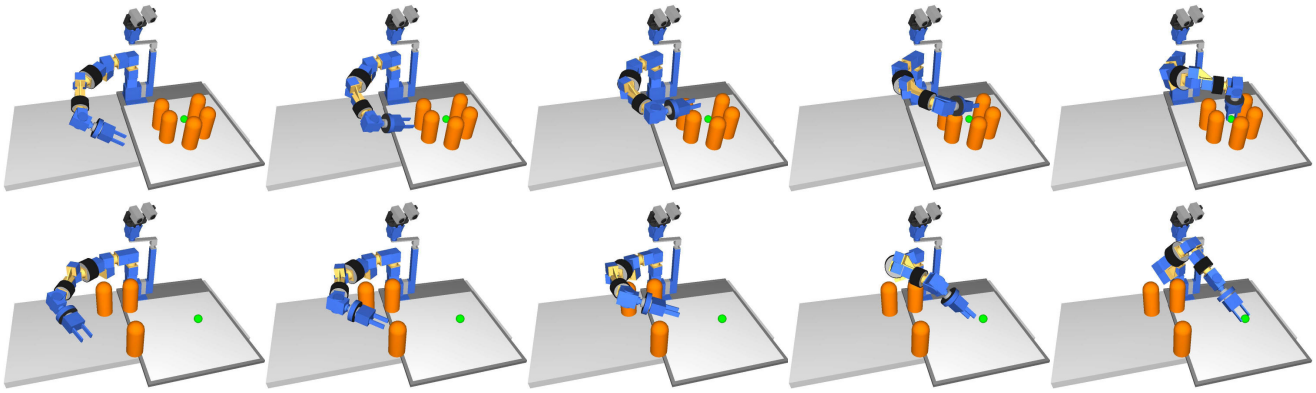

Fig. 6. Two sequences of simulated movements of CoRA reaching for a target. The obstacles were manually arranged to pose interesting challenges.

TABLE I

RESULTS OF EXPERIMENTS WITH RANDOMIZED SCENES

No. of obst.	N	success (%)	failure (%)	Failure reasons (%)				
				P	W	O	X	D
1	1000	99.7	0.3	0.1	0.2	-	-	-
3	1000	99.1	0.9	0.7	0.1	-	0.1	-
6	1000	95.4	4.6	2.5	0.3	0.9	0.7	0.9
10	1000	92.7	7.3	3.5	0.5	2.0	0.7	0.6
15	500	87.0	13.0	6.4	0.4	4.6	1.2	0.4
20	200	83.0	17.0	7.0	-	8.0	1.5	0.5

with a large obstacle near the first link segments, but chosen in a way that the manipulator has barely enough freedom to find a path above it (fig. 6, lower panel). In both cases the behavior generation scheme finds viable trajectories that reach the target and avoid obstacle collision.

IV. CONCLUSIONS AND FUTURE WORK

We presented a general scheme for the autonomous generation of collision free reaching movements for arbitrary redundant manipulators. Multiple constraints from the target and different obstacles were formulated as attractors or repellors in locally relevant variables and then combined by transforming them into a vector field over the joint velocities. As a demonstration the system was implemented for an 8 DoF robotic arm. Extensive experiments showed the capability of the behavioral scheme to successfully find paths in a wide array of situations, some of them through very cluttered scenes.

Shortcomings revealed by the experiments include a cancellation effect in which contributions from different obstacles push the joint configuration into opposite directions in joint space, effectively weakening the avoidance effect. This may allow the manipulator to come very close to the obstacle surfaces. The associated dynamic contributions from obstacle avoidance then becomes very large and the overall dynamical system becomes numerically unstable. A second source of error is insufficient damping of velocities in the task null-space that may prevent the system from finding valid trajectories toward the movement target. Both errors occurred only in a small numbers of cases. A more severe downside is the disability to find paths that require significant configuration changes, but this is a general disadvantage of local approaches to movement generation. While it should be added that many successful paths did realize such configuration changes, there is definitely room for improvement in this matter, and all of these points are directions of future refinements of our system.

Finally, the relatively large number of parameters that play a role in the dynamical system may be problematic. For the present work, these parameters were tuned by hand, depending on the experience and insight of the designers. In the past, we have successfully used automatic parameter adaptation through evolutionary optimization in related mathematical settings [22]. These methods will be applied to the dynamical approach to arm movement system in future work.

Overall, more work is still necessary to examine the robustness of the system and its applicability to real world scenarios. Although the problems of joint angle limits and of constraints on velocities, accelerations and torques have not been attended to yet, solutions within the framework of attractor dynamics are conceivable, including setting repellors in the directions in joint velocity space that point to the undesired values.

Other areas of future work include different types of obstacles, e.g. surfaces, objects located higher up in the workspace, and self-collision of link segments with each other. Also, moving obstacles will be used to demonstrate the capability of the scheme to generate behavior in dynamic environments. So far only reaching was modelled, because the focus lay on the obstacle avoidance part. In order to be usable in real world scenarios, we will include actual grasping of objects. A key aspect for this is achieving a desired orientation of the hand relative to the target, which will be formulated as an attractor in orientation space to be consistent with the dynamic systems approach. Another factor is the shape of the generated trajectories, which tend to turn away from obstacles rather sharply. This is a disadvantage for collaboration between humans and robots in a shared environment, as it makes prediction of the manipulator movements difficult for humans.

V. PARAMETERS

A. Target acquisition

$\alpha_\phi = 10$; $v_{des} = 150\,\text{mm}/\,\text{sec}$; $\alpha_{vel} = 15$;
$\alpha_p = 5$; $\alpha_v = 25$; $\alpha_{damp} = 10$;
$d_1 = 5\,\text{mm}$; $d_2 = 15\,\text{mm}$

B. Obstacle avoidance

$\delta_1 = 15\,\text{mm}$; $\delta_2 = 50\,\text{mm}$;
$\psi_1 = 0.25\,\text{rad}$; $\psi_2 = 1.5\,\text{rad}$; $\alpha_{obs} = 50$

C. Implementation and experiments

computation cycle length: 25ms
$\theta_{init} = \big(0.2,\ 0.0,\ 0.4,\ 0.0,\ 0.5,\ -0.3,\ 1.0,\ 0.0\big)\,\text{rad}$;
$\sigma_i = 0.1\,\text{rad}^2, 1 \leq i \leq 8$
target area center: $\big(-250,\ 450,\ 250\big)\,\text{mm}$
target area extensions: $\big(500,\ 700,\ 400\big)\,\text{mm}$
obstacle area center: $\big(-100,\ 550,\ 0\big)\,\text{mm}$
obstacle area extensions: $\big(800,\ 700,\ 0\big)\,\text{mm}$
obstacle radius: 35–60mm
obstacle height: 100–400mm

ACKNOWLEDGMENT

The authors acknowledge support from the German Federal Ministry of Education and Research within the National Network Computational Neuroscience – Bernstein Fokus: "Learning behavioral models: From human experiment to technical assistance", grant FKZ 01GQ0951.

REFERENCES

[1] G. Schöner, M. Dose, and C. Engels, "Dynamics of behavior: Theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, vol. 16, pp. 213–245, 1995.

[2] E. Bicho and G. Schöner, "The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform," *Robotics and autonomous systems*, vol. 21, pp. 23–35, 1997.

[3] I. Iossifidis and G. Schöner, "Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 580–585.

[4] B. R. Fajen, W. H. Warren, S. Temizer, and L. P. Kaebling, "A dynamical model of visually-guided steering, obstacle avoidance and route selection," *International Journal of Computer Vision*, vol. 54, no. 1-2, pp. 13–34, 2003.

[5] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[6] H. Choset, S. Hutchinson, K. Lynch, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. The MIT Press, 2005.

[7] B. Burns and O. Brock, "Toward optimal configuration space sampling," in *Robotics: Science and Systems (RSS)*, 2005.

[8] R. B. Rusu, I. A. Sucan, B. Gerkey, S. Chitta, M. Beetz, and L. E. Kavraki, "Real-time Perception-Guided Motion Planning for a Personal Robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA*, 2009.

[9] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, p. 90, 1986.

[10] J. R. Andrews and N. Hogan, "Impedance control as a framework for implementing obstacle avoidance in a manipulator," in *Control of Manufacturing Processes and Robotic Systems*, E. E. Hardt and W. Book, Eds. ASME, Boston, 1983, pp. 243–251.

[11] O. Brock and O. Khatib, "Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths," in *International Conference on Robotics & Automation San Francisco, CA*, 2000.

[12] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, pp. 505–518, 2005.

[13] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The international journal of robotics research*, vol. 4, no. 3, p. 109, 1985.

[14] A. McLean and S. Cameron, "The virtual springs method: path planning and collision avoidance for redundant manipulators," *The International Journal of Robotics Research*, vol. 15, pp. 300–319, 1996.

[15] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[16] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," *Advances in neural information processing systems*, pp. 1547–1554, 2003.

[17] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *International Conference on Humanoid Robots*, 2008.

[18] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Proceedings of the IEEE/RAS International Conference on Robotics and Automation (ICRA)*, 2009.

[19] F. Stulp, E. Oztop, P. Pastor, M. Beetz, and S. Schaal, "Compact models of motor primitive variations for predictable reaching and obstacle avoidance," in *9th IEEE-RAS International Conference on Humanoid Robots*, 2009.

[20] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Transactions on Robotics and Automation*, vol. 3, pp. 308–316, 1987.

[21] I. Iossifidis, C. Bruckhoff, C. Theis, C. Grote, C. Faubel, and G. Schöner, "Cora: An anthropomorphic robot assistant for human environment," in *Robot and Human Interactive Communication*, 2002, pp. 392–398.

[22] A. Pellecchia, C. Igel, J. Edelbrunner, and G. Schöner, "Making driver modelling attractive," *IEEE Intelligent Systems*, vol. 20, pp. 8–12, 2005.