The Bellmann Mapping Machine for Nonlinear Approximation in Control Policy Space

G. Fahler, R. Eckmiller
University of Bonn
Dept. of Computer Science VI (Neuroinformatics)
FR Germany
Tel.: ++49-228-550-364 FAX: ++49-228-550-425 e-mail: gerald@snor.uni-bonn.de

Abstract We propose a novel scheme, named 'Bellmann Mapping Machine' (BMM), that aims to extend the scope of reactive robot controllers towards more complex tasks. BMMs are implemented by shallow feed forward networks, that receive as input the compound information about desired actions, task, present robot state, and short-term predicted cluttered constraints. The street-crossing problem serves as a test-bed for our performance studies: the task there is to generate optimal goal-directed robot motor trajectories, that avoid collisions with moving obstacles, at the same time respecting the robot's dynamics limitations. We supervise some novel higher order recurren controller with optimal control examples as computed by Dynamic Programming. We find very efficient representations of the underlying optimal control policy space, as well as sensible generalization to new control situations.

1. Introduction Flexible and robust control of robots acting in rapidly changing environments must give special emphasis to real-time sensorimotor integration. Computationally shallow, inexpensive reactive system designs are preferable over iterative global search methods, particularly when robot position constraints become time-variant in the presence of moving obstacles. A branch of reactive system designs suited for obstacle avoidance emanates from the idea of furnishing Euclidean space with spatial potential fields [5]. Along with the representational simplicity of these schemes there go however limitations, such as emergence of local minima for entangled obstacle constellations. In addition, spatial field representations fail to take into account dynamics limitations of the robot, and lack temporal reasoning qualities. Every-day experience in crossing busy streets, as an example, makes obvious, that overcoming these representational inadequacies would distinctly enhance the survivability of inert robots exposed to speedy obstacles.

A profound demand on robot control is formulated as the 'kinodynamic path planning problem' [5]: a given a robot system, and some desired goal, find a cost-optimal trajectory, that avoids moving obstacles, while respecting bounds on robot velocities and accelerations. Can this formidable task be handled by shallow circuits at all? And are feed forward nets powerful enough to represent the underlying control policy spaces? Or does the temporal dimension of the task demand on relaxation-like controller designs, equipped with internal feedback loops and sequential processing capabilities [6, 7]?

In this paper we try to give partial answers to these questions, largely based on simulation results. For the task discussed above, we implement the BMM as an adaptively structured, nonlinear feed forward neural net that serves as an approximator of the optimal control policy mapping. We investigate the capacity of these distributed, and sparse representations, to generalize from control rule examples. In section 2 of this paper we introduce robot and environment models. The proposed sensorimotor system design and functionality is described in section 3. Section 4 discusses our method to obtain optimal control examples by means of Dynamic Programming (DP). The neurocontroller model is then introduced. In section 5, results of extensive simulations are reported. The paper concludes with a discussion of the capabilities and limitations of the proposed approach.

2. Environment and Robot Models The world around the robot is a two-dimensional scene, occupied by square obstacles moving all in parallel to the p-axis, with randomly chosen discretized p-positions, and with a continuous velocity spectrum. The environment's state is given by a list
reporting position, and velocity of each obstacle $i$. The environment dynamics is given by

$$y_i(t+1) = y_i(t) + u_i.$$  

A point-like robot of unit mass confined to some interval along the $x$-axis, and obeys a discretized position-/velocity spectrum: $X \in \{0, ..., 8\}$; $X = \{-1, 0, 1\}$. At each time step, a motor command $u \in X = \{-1, 0, 1\}$ is applied to the robot. Dynamic equations are given by

$$\begin{align*}
\dot{x}_r(t+1) &= \dot{x}_r(t) + u(t) \\
x_r(t+1) &= x_r(t) + \dot{x}_r(t+1).
\end{align*}$$  

Notice that the set of admissible motor commands depends on the present robot state.

The above setting for our test-bed sketch a robot with limited dynamics, which has to plan ahead in time, in order to avoid fluctuating numbers of obstacles, that cross its baseline in ever new constellations. The situation is similar to that of a pedestrian crossing a busy street (Figure 1).

3. Sensorimotor System Design and Functionality

BMM receives as input the compound information about desired action task, present robot state, and cluttered spatiotemporal constraints as extending over some limited future time interval (Figure 2). The latter information is not immediately available from the environment. In [4], a pre-processing device, denoted there as Perception Module, that implements the necessary environment model, was discussed. From sensory information it computes short-term forecasts of future obstacle positions. In effect, it assembles some robo-centric constraints vector, whose the components label the occupancy state of those spacetime cells, that are within reach for the robot within a finite planning horizon.

BMM acts as an inverse model within a closed-loop control cycle. It generates state-dependent robot motor accelerations, that aim to move the robot towards the desired goal position as specified by the action task input, while respecting the constraints vector.

4. Computation and Approximation of Bellman Machines

Firstly, we realize computation of the optimal control policy by DP [1]. Secondly, we use supervised learning to distribute examples of this ‘Bellman mapping’ over the neural BMM.

At every timestep $t$, DP determines a sequence of motor commands minimizing some cost functional. Here we use the quadratic finite-horizon version:

$$\text{cost}(u(t), \ldots, u(t+\text{HORIZON})) = \sum_{k=0}^{\text{HORIZON}} (x_r(t+k) - x^o)^2 + c u(t+k)^2,$$

with $x_r(t+k)$ given by the dynamics eqns.(2). By $x^o$, we denote the desired robot position. Deviations from this, as well as costly accelerations, are punished by higher costs. Collision-free
The environment dynamics is given by:

$$y(t) = u(t) + v(t)$$

(1)

where $v(t)$ is the external noise and $u(t)$ is the control input.

The goal is to design a controller that minimizes the cost function $J(u)$, which is given by:

$$J(u) = \frac{1}{2} \int_{t_0}^{t_f} [y(t) - r(t)]^2 dt$$

(2)

where $y(t)$ is the system output, $r(t)$ is the reference input, and $t_0$, $t_f$ are the initial and final times, respectively.

5. Simulation Results

The controller was evaluated in a simulated environment to test its effectiveness. The simulation was performed using a robot model with known dynamics and a desired trajectory $r(t)$. The controller was designed to minimize the cost function $J(u)$.

The robot dynamics are described by the following differential equation:

$$\dot{x}(t) = f(x(t), u(t), t)$$

(3)

where $x(t)$ is the state vector, $f$ is the nonlinear function describing the system dynamics.

The control input $u(t)$ is designed to track the desired trajectory $r(t)$ and minimize the cost function $J(u)$. The controller design incorporates a feedforward term $u_{ff}(t)$ and feedback terms $u_{fb}(t)$, which are calculated based on the error $e(t) = r(t) - y(t)$.

$$u(t) = u_{ff}(t) + u_{fb}(t)$$

(4)

The feedback terms are calculated using a feedforward neural network (FFNN) that maps the error $e(t)$ to the control input $u_{fb}(t)$.

$$u_{fb}(t) = \text{FFNN}(e(t))$$

(5)

The feedforward network is trained using historical data to predict the control input for a given error. The training process involves minimizing the cost function $J(u)$ using an optimization algorithm such as gradient descent.

The simulation results show that the controller is able to accurately track the desired trajectory and minimize the cost function $J(u)$. The performance of the controller is evaluated using various metrics, including tracking error, control effort, and computational efficiency.

6. Conclusion

The results demonstrate the effectiveness of the proposed controller in tracking a desired trajectory and minimizing the cost function. The controller design incorporates both feedforward and feedback terms to achieve good performance in dynamic environments.

The controller can be further improved by incorporating adaptive learning algorithms to adjust the weights of the neural network based on the performance. Additionally, the controller can be extended to handle more complex scenarios with multiple robots and obstacles.
exhibits many terms of orders 4, 5, 6, and higher, and finally decreases to zero for orders exceeding 10. These findings hint on the high-order character of the Bellman mapping for the kinodynamic path planning problem.

5. Conclusions Sparse representation of control laws is desirable when table look-up becomes impracticable (Bellman's 'curse of dimensionality'), and when iterative computation of optimal policies becomes too expensive, or conflicting with real-time requirements. Some mechanism of generalization, which turns already acquired control skills over to new task instances, can distinctly improve the survivability of sensory driven robots. For these reasons it is urgent to investigate the competence of neurocontrol for efficient distributed representation, and for robust generalization of optimal control policies.

Here, we focused on a new type of shallow feed forward neurocontroller for local kinodynamic trajectory planning. An advantage with feed forward nets is their low-latency recall, and their relatively quick learning, as compared to recurrent networks. However, from theoretical considerations concerning the non-local nature of the related connectedness predicate [6], the problem under focus is expected to be hard for feed forward nets, when scaled up. Even for limited time-horizons, complex, nonlinear, and jumplike optimal control policies must be faced, due to constraint-induced bifurcations of optimal phase-space trajectory bundles. We met the required mapping complexity with a powerful novel classifier model supporting effective computation, and automatic identification, of the relevant nonlinearities inherent in the mapping. We found extremely parsimonious distributed representations of optimal control policies, indicating that some compact set of important high-order features determines the optimal control. The neural BMMs emerged excellent generalization to new control task encounters.

We encourage use of feed forward neurocontrol for approximation of Bellman mappings obeying cluttered constraints, but care must be taken that the models support efficient representation of high-order nonlinearities. For growing time-horizons, it is expected that feed-forward neurocontrol will run into limitations [7]. Some deficiency of our approach is its burden with increasing constraints vector dimension for growing planning-horizons. This objection is weakened, however, when considering partially unmodelled natural environments, where long-term planning is not feasible due the absence of globally disposable constraints.

References


