Estimation of Obstacle-Aware Reachability Maps for Arbitrary Three-Link Planar Manipulators Using one Surrogate Model

Simon Schläger, Stefan Groß, Hailong Luo, Mathias Hüsing, and Burkhard Corves

Abstract A surrogate model that takes into account the kinematic parameters of a three-link planar manipulator is presented for estimating obstacle-aware reachability maps of that manipulator. In contrast to prior work, the proposed model jointly conditions on occupancy grids and kinematic parameters, enabling reuse across different manipulators and supporting kinematic design and optimization tasks. A U-Net-based encoder-decoder network with a dedicated parameter branch is trained on a large synthetic dataset with random manipulators and cluttered environments. For a set of kinematic parameters seen during the training, the proposed surrogate model achieves a significantly lower prediction error and faster inference than the current baseline. Furthermore, the surrogate model is able to generalize well over unseen kinematic parameter sets and obstacle layouts. The results prove the concept for fast reachability estimation that could be extended to SE3 space for design optimization, workspace analysis and pathplanning in future works.

S. Schläger
RWTH Aachen Univerity, Aachen 52062, Germany
e-mail: schlaeger@igmr.rwth-aachen.de

S. Groß
RWTH Aachen Univerity, Aachen 52062, Germany

H. Luo
RWTH Aachen Univerity, Aachen 52062, Germany

M. Hüsing
RWTH Aachen Univerity, Aachen 52062, Germany

B. Corves
RWTH Aachen Univerity, Aachen 52062, Germany

1 Introduction

Robot manipulators are a cornerstone of modern automation. In the past, industrial robotic arms were fixed to the ground and typically worked in structured cells, which simplified planning but limited the achievable working range. The rise of mobile manipulators expands this limited workspace. However, this introduces a new challenge for planning: the mobile base must be positioned so that the end effector can reach and align with the target poses relevant to the task in cluttered, dynamic scenes. When selecting the basic position of the manipulator, it is therefore essential to have precise, task-specific descriptions of its performance capabilities, which are usually presented in the form of reachability maps of the workspaces. [3]

Zacharias et al.[8] introduced reachability maps that measure the number of inverse kinematic solutions found for each discretized subsection of the workspace. Vahrenkamp et al. [6] extended this idea by incorporating joint limits and obstacles. Instead of a ration of reachable poses Vahrenkamp et al. used the Yoshikawa [7] manipulability measure as a quality index for each discretized subsection of the workspace. Both methods require extensive computing time, which is why they can usually only be precalculated offline. Sandakalum et al.[5] showed that such offline created maps can be poor approximations in cluttered dynamic scenes, because the influence of an obstacle extends beyond its physical boundaries and alters the reachability of neighboring regions. To obtain real-time capable estimates, Sandakalum et al. [5] proposed a data-driven surrogate model that predicts the obstacle-aware reachability map of a three-link planar manipulator directly from an occupancy grid map representation of the world.

The surrogate model proposed by Sandakalum et al. [5] has one major limitation: it does not explicitly include the kinematic parameters of three-link planar manipulator and is therefore specific to a single manipulator. As a consequence, a new surrogate model must be trained for each individually parameterized manipulator, which is particularly costly in SE(3) space. In addition, the lack of an explicit dependence on kinematic parameters prevents the use of the model in kinematic design optimization tasks [2]. This work closes this gap by introducing a surrogate model that takes into account both obstacles in the workspace and the kinematic parameters of three-bar planar manipulators.

2 Background of Reachability Maps

According to Zacharias et al. [8], reachability maps describe the workspace of a manipulator. Reachability maps are obtained by discretizing the robot workspace into voxels and testing how many sampled end effector poses can be realized in each voxel. To do this, a sphere is created at each voxel center with a diameter equal to the voxel edge length, see fig. 1. A set of N end effector poses is sampled on the surface of this sphere. The inverse kinematics problem is solved for each sampled pose. If there is a feasible solution, it is counted towards the number of end effector poses R. To obtain a realistic map, each possible configuration can also be checked for collisions and self-collisions. A colliding pose then becomes unfeasible, even though a solution exists in principle. Off-the-shelf collision detection libraries can be used for this purpose. Finally, the reachability index D of a voxel is defined as the ratio between the number of feasible poses R and the number of sampled poses N per voxel:

$$D = \frac{R}{N}, \tag{1}$$

Usually, the origin of the voxel space lies within the origin of the robot base. The voxel edge length directly defines the spatial resolution of the map. With higher spatial resolution, the computing time increases dramatically.[3]

To enable online calculation, Sandakalum et al. [5] introduced a surrogate model that receives a 2D occupancy grid as an input and returns the corresponding obstacle-aware reachability map of a three-link planar manipulator. The surrogate model is implemented as a neural network, whereby both the occupancy map as input and the reachability map as output are passed to the network or returned from the network as one-dimensional arrays. The architecture of the surrogate models consists of three fully connected hidden layers with 2000, 4000, and 2000 neurons, respectively. All layers employ ReLU activation functions. This surrogate model, referred to as ReachNet [5], can estimate reachability maps at about 33 Hz and therefore provides obstacle-sensitive reachability map at a rate that is suitable for integration into pathplanning algorithms. [5]

3 Methodology

The following section presents a method for constructing a data-driven surrogate model that jointly accounts for workspace obstacles and the kinematic parameters of a three-link planar manipulator with three rotational joints.

3.1 Data Generation for Training and Validation Datasets

To train the data-driven surrogate model, first a large dataset of reachability maps for planar three-link manipulators is generated. For this purpose, the kinematic parameters of the planar three-joint manipulators were generated by randomly selecting the joint lengths from a predefined interval 12 to 21 length units and assigning a symmetric range of motion to each joint. The value for the symmetrical joint range is between $\pm 120°$ and $\pm 180°$. Each manipulator can therefore be described by six parameters, namely three link lengths and three symmetric joint-limit values. In total, 128 distinct three-link planar manipulators are created to obtain a diverse set of kinematic configurations for learning the surrogate model.

For every robot, a set of two-dimensional occupancy grid maps, that represent different obstacle arrangements in the workspace, is constructed. The grid resolution was fixed at 48×48 pixels and for each robot 1280 different maps were generated, each containing a random number of two to six obstacles with randomly sampled sizes. The continuous workspace of each robot was discretized to the 2D pixel grid such that the entire manipulator fits into the map. This implies that larger manipulators are represented with a coarser length resolution per pixel than smaller ones.

In a next step, the corresponding reachability maps are determined for all randomly generated occupancy maps, taking into account the kinematic parameters of the respective planar manipulator. To this end, the method of Zacharias et al. [8] is adapted to the planar case. As Fig. 1 showcases, each pixel was treated as a 2D analogue of a voxel, with a circle defined at the center of each pixel. On the circumference of the circle, 100 end effector poses with different orientations are sampled in a uniform, deterministic manner. For each sampled pose, the inverse kinematics problem is solved and the resulting configuration was checked for collisions with obstacles, while self-collisions are ignored under the assumption that planar manipulators with suitably spaced links can be designed to be effectively free of self-collision. This procedure yielded one reachability map per occupancy grid, providing the paired data that links occupancy maps to their corresponding obstacle-influenced reachability distributions.

To benchmark the proposed approach, an additional separate dataset is generated for a single three-link planar robot with 10240 occupancy maps, which is a similar amount of data to the state of the art [5]. The manipulator's kinematic parameters were taken from the reference work [5]. This single-robot dataset provides a controlled setting that allows a direct and fair comparison with existing surrogate models from the literature.

3.2 Network Architecture

To learn the mapping from occupancy grids and robot parameters to reachability maps, a convolutional encoder–decoder architecture of U-Net [4] type with an additional parameter branch is used. The schematic overview of the network is given in fig. 2. The first input is a single-channel 48×48 occupancy map that is processed by a sequence of convolutional blocks into a contracted feature map. The second input is the kinematic parameters of a three-link planar manipulator (link lengths and joint constraints), which are

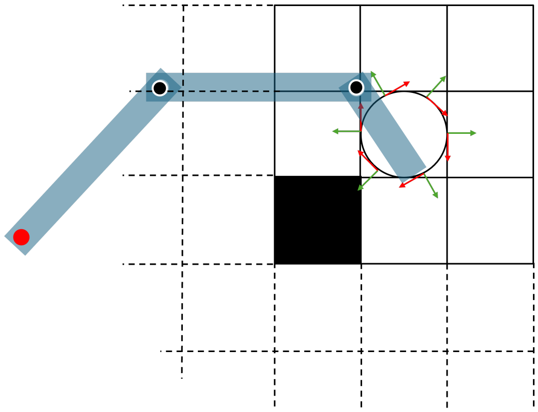

Fig. 1 Representation of the pose sample processes for one enlarged pixel: A circle is placed at the pixel center. The sampled poses on the circumference are defined such that the y-unit vector is normal and the x-unit vector is tangential to the circumference path. Black pixels represent the occupied pixel in the occupancy grid map.

fed into a parallel separate multilayer perceptron that maps the parameter vector to a fixed-dimension embedding. At the bottleneck of the U-Net encoder, this embedding is concatenated with the convolutional feature maps, and further processed by convolutional block to fuse spatial and parametric information. The U-Net decoder then reconstructs a dense prediction at the original resolution using convolutions for upsampling and skip connections from the encoder. This architecture thus combines local convolutional processing of the occupancy grid with a global conditioning on the robot parameters to predict obstacle-aware reachability maps.

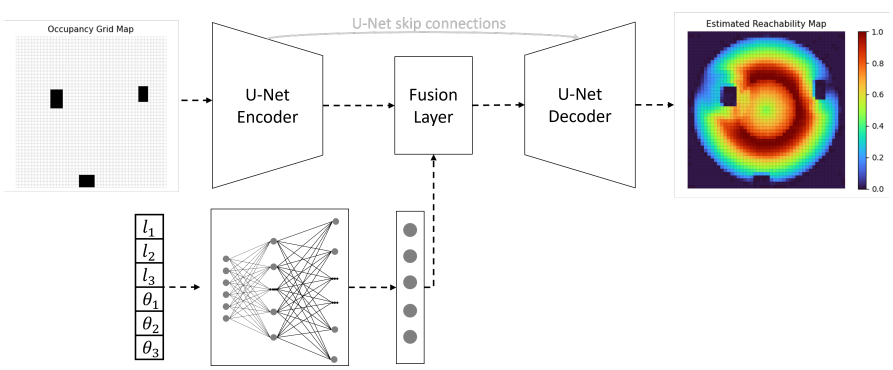

Fig. 2 Schematic overview of the parameter-conditioned U-Net with bottleneck fusion: On the left hand the occupancy grid map and the kinematic parameters link lengths l_1 to l_3 and symmetric joint limits θ_1 to θ_3 are given as input. On the right hand the estimated reachability map is depicted.

3.3 Training and Validation

To provide a benchmark against Sandakalum [5] and to evaluate the surrogate model extended by kinematic parameters, two separate surrogate networks are trained that use the same architecture and identical training hyperparameters, but two different datasets mentioned in section 3.1. Both surrogate models are trained on a Windows terminal server equipped with a NVIDIA® RTX 4060 Ti 16GB GPU and a 24 core AMD® Ryzon Threadripper PRO CPU with a base speed of 3,80 GHz. As the loss function PyTorch's [1] SmoothL1Loss is used. The surrogate models are trained with the Adam optimizer for 20 epochs with a learning rate of 1×10^{-4} and a batch size of eight. The total combined training time is about one and a half hours. The hyperparameters were chosen based on a few preliminary experiments rather than an extensive hyperparameter study, as even this minor adjustment leads to strong performance. Both datasets are roughly split into 80 % for training and 20 % for validation, where the validation set for the variable kinematic parameters contains both previously unseen robots and new reachability maps for robots present in the training set. Details can be found in table 1. As a validation metric, the mean squared error (MSE) was chosen, as this is also the used metric in the work of Sandkalum et al. [5].

Table 1 Training and validation splits for the two conducted tests: The numbers are based on 32, which is the selected number of maps per stored file.

	Number of Robots		Number of Maps per Robot		Total Maps
	Train	Validation	Train	Validation	
Benchmark	1	-	8192	2048	10240
Our Approach	96	32	768	256	131072
					141312

4 Results

This section is divided into two parts. First, we benchmark the proposed surrogate model against the current state of the art, represented by the ReachNet approach of Sandakalum [5]. To improve comparability and counteract the performance boost of current hardware, all inference times listed below were measured on a weaker test machine that only uses an Intel Core i7-1360P CPU with a base clock speed of 2.2 GHz and no GPU acceleration. Therefore, the reported runtimes represent a conservative estimate for modern practical applications. Second, results for the extended setting with varying kinematic configurations are presented, and the generalization capabilities of the proposed method across different three-link planar manipulators are analyzed.

4.1 Evaluation for a Fixed Single Kinematic Parameter Set

This subsection compares our proposal to the benchmark provided by Sandakalum et al. [5]. To ensure a fair comparison, a data set of equal size consisting of 10,240 occupancy maps (2048 validation sets) is used with a single robot. In addition, the kinematic parameters are set to the values specified by Sandakalum et al. [5]. In the original work, a MSE of 1.078×10^{-3} was reported with an inference time of 30 ms on an NVIDIA® GTX 1080 GPU. Our proposed method achieves a MSE of 2.31×10^{-4} with an average inference time of 6 ms on our described inference system, demonstrating better accuracy at a reduced prediction time. Together, the results show that the proposed method outperforms the benchmark by Sandaklaum et al. [5]. Also, it becomes clear that convolutional encoder–decoder models provide a viable and effective alternative to fully connected surrogate models for learning obstacle-aware reachability maps.

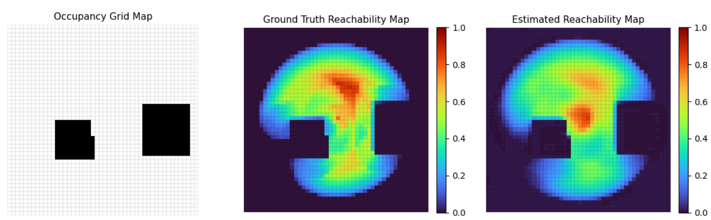

Fig. 3 From left to right: occupancy grid map with random obstacles, ground truth reachability map based on analytical computation, predicted reachability map based on surrogate model.

4.2 Evaluation for Arbitrary Multiple Kinematic Parameter Sets

Based on the created datasets in section 3.1, two different types of validation measures can be evaluated. The first type is computed for manipulators whose kinematics parameters are known from training but which are evaluated on previously unseen occupancy grid maps. This setting corresponds to the benchmark with a single manipulator, but the network must now simultaneously handle multiple kinematic parameter sets. The second type of validation error is obtained for kinematic parameter sets that were not included in the training set and are evaluated in previously unseen environments, so that both the kinematics and the obstacle configurations are new to the surrogate model. Together, these two measures separate generalization over environments from generalization over manipulators' kinematic parameters.

In the first experiment, the validation error is evaluated for a manipulator whose kinematic parameters are contained in the training set, but which is queried with previously unseen occupancy maps. An MSE of 1.68×10^{-4} is obtained, which is even lower than in the case with fixed single kinematic parameter set, indicating that obstacle-aware reachability maps can be reliably predicted for manipulators matching the training kinematics. In a second experiment, the model is validated by applying it to occupancy grid maps and kinematic parameter sets that the model did not observe during training This yields an MSE of 2.9×10^{-3}, which remains in the region of Sandakalum's et al. [5] reported error and shows that generalization beyond the training set to new three-link robots with different link lengths and joint limits is achieved. The validation errors for both cases are summarized in table 2, thereby contrasting generalization over environments with generalization over robot kinematics.

Table 2 Results for arbitrary multiple kinematic parameter sets: Given are the mean values of all MSE from the respective validation sets and their standard deviation (STD).

	Unseen Kinematic Set		Seen Kinematic Set	
	Mean	STD	Mean	STD
SmouthL1Loss [$\times 10^{-4}$]	13	18	0.84	1.1
MSE [$\times 10^{-4}$]	29	37	1.68	2.2
Inference time [ms]	6.7	1.2	6.4	1.0

5 Discussion & Future Potential

This work interprets the results mainly as a proof of concept in SE(2) space, where surrogate models bring limited practical benefit because reachability maps for planar manipulators are already cheap to compute. The approach becomes relevant in SE(3) space, where generating high-resolution reachability maps for spatial manipulators is computationally expensive and can become a bottleneck for many robot variants or frequent updates. Surrogate models operating on voxelized occupancy data and robot parameters could significantly accelerate this process while maintaining sufficient accuracy for design and planning.

Promising SE(3) applications include (i) supporting kinematic design optimization and robot cell layout optimization, and (ii) serving as fast reachability estimators for positioning the mobile base in path planning pipelines for dynamic and clustered environments. Future work should extend the framework to other map-based performance measures (e.g. manipulability, dexterity, task-specific indices) and incorporate connectivity or path feasibility, rather than only static pose feasibility, to better capture practical reachability in cluttered workspaces.

Acknowledgements Part of the research presented was funded by the Europen Union as part of the SHEREC research project. [Grant agreement ID: 101136056]

References

1. PyTorch documentation — PyTorch 2.9 documentation. URL https://docs.pytorch.org/docs/stable/index.html
2. Külz, J., Mayer, M., Althoff, M.: Timor python: A toolbox for industrial modular robotics. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 424–431. DOI 10.1109/IROS55552.2023.10341935. URL https://ieeexplore.ieee.org/abstract/document/10341935. ISSN: 2153-0866
3. Makhal, A., Goins, A.K.: Reuleaux: Robot base placement by reachability analysis. In: 2018 Second IEEE International Conference on Robotic Computing (IRC), pp. 137–142. DOI 10.1109/IRC.2018.00028. URL https://ieeexplore.ieee.org/document/8329892
4. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. DOI 10.48550/arXiv.1505.04597. URL http://arxiv.org/abs/1505.04597
5. Sandakalum, T., Ang, M.H.: ReachNet : Reachability maps in the presence of obstacles. In: 2023 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), pp. 108–113. DOI 10.1109/CIS-RAM55796.2023.10370757. URL https://ieeexplore.ieee.org/abstract/document/10370757. ISSN: 2326-8239
6. Vahrenkamp, N., Asfour, T.: Representing the robot's workspace through constrained manipulability analysis **38**(1), 17–30. DOI 10.1007/s10514-014-9394-z. URL https://doi.org/10.1007/s10514-014-9394-z
7. Yoshikawa, T.: Manipulability and redundancy control of robotic mechanisms. In: 1985 IEEE International Conference on Robotics and Automation Proceedings, vol. 2, pp. 1004–1009. DOI 10.1109/ROBOT.1985.1087283. URL https://ieeexplore.ieee.org/document/1087283
8. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: representing robot capabilities. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3229–3236. DOI 10.1109/IROS.2007.4399105. URL https://ieeexplore.ieee.org/document/4399105. ISSN: 2153-0866