Manipulator-Based Grasping Pose Selection by means of Task-Objective Optimisation

David Richards, Gavin Paul, Stephen Webb, Nathan Kirchner
ARC Centre for Autonomous Systems
University of Technology, Sydney, Australia
{david.richards, gavin.paul, stephen.webb, nathan.kirchner}@uts.edu.au

Abstract

This paper presents an alternative to inverse kinematics for mobile manipulator grasp pose selection which integrates obstacle avoidance and joint limit checking into the pose selection process. Given the Cartesian coordinates of an object in 3D space and its normal vector, end-effector pose objectives including collision checking and joint limit checks are used to create a series of cost functions based on sigmoid functions. These functions are optimised using Levenberg-Marquardt's algorithm to determine a valid pose for a given object. The proposed method has been shown to extend the workspace of the manipulator, eliminating the need for precomputed grasp sets and post pose selection collision checking and joint limit checks. This method has been successfully used on a 6 DOF manipulator both in simulation and in the real world environment.

1 Related Work

The growing demand for robots to support the independence of humans in their homes is fuelled largely by the ageing population and the increasing cost of health care. The ability of robots to manipulate objects is a vital capability in the real world, for example being able to pick up and move common household objects. The home environment is dynamic and may be initially unknown, precluding the use of scripting and tele-operating methods of object grasping [Shin-ichi and Satoshi, 2000]. Instead, this problem predicates an intelligent method for object grasping that is capable of adapting to a variety of scenarios commonly experienced in the home.

Autonomous methods of object manipulation have two important components: identifying a valid grasp pose and planning a path for the end-effector from the current position to the grasp pose, as well as the final approach. Much work has already been done in the path planning component of the grasp problem [Yakey *et al.*, 2001; Kuffner and LaValle, 2000]. However, the area of grasp pose selection continues to be a challenging and complex problem. Past research shows that inverse kinematics is a widely used and accepted solution to pose selection due to its ability to calculate grasp poses for robotic manipulators [Bicchi and Kumar, 2000; Miller *et al.*, 2003; Pelossof *et al.*, 2004; Hsiao and Lozano-Perez, 2006; Berenson *et al.*, 2007; 2008; Bertram *et al.*, 2006; Mason, 2001]. There are several challenges in using inverse kinematics, including the presence of singularities in pose configurations [Dinesh K. Pai, 1992], the common need to generate an extensive offline database of potential grasps to increase the online computational speed, and the lack of simultaneous collision checking and joint limit checks.

A common method for solving the grasp pose selection singularity problem is to use an iterative, numerical technique of inverse kinematics based on pseudo-inverse of the Jacobian J+ calculations [Klein and Huang, 1983; D'Souza *et al.*, 2001]. Although the corresponding pseudo inverse can be defined for a singular matrix, the solution is discontinuously switched from an exact one to an approximate one at singular points. Cheng *et. al.* [1995] describe an approach to solve this issue with the Singularity Isolation plus Compact QP (SICQP) method. The SICQP method decomposes the work space into achievable and unachievable (i.e. singular) directions. This method retains the exactness of the solution, however it relies on a segregated workspace which limits the space where the manipulator can move.

As well as the singularity issue, inverse kinematic solutions are often reliant upon large grasp sets being developed off-line and then checked online. A common framework to develop this is using force-closure [Mason, 2001]. This method involves sampling the grasp

parameter space to find a set of grasps for the object of interest [Morales *et al.*, 2006]. Many of the solutions in this set are invalid due to manipulator collisions with the environment and poses that are unreachable because of the kinematic limitations of the arm. These solutions must be discarded from the pose set after checking. A näive method to find a valid solution would be to check the solutions randomly until a valid pose is found. However, due to the potentially large number of grasps in a grasp set, there is a trade-off between computational efficiency and workspace size. Increasing the workspace also increases the size of the grasp set and the time taken to find a valid pose. This means that a manipulator workspace must be reduced to a selected area to increase the efficiency of the pose selection.

Berenson *et. al.* [2007] address the issue of computational efficiency by developing a grasp scoring function to rank the set of poses that combine a grasp quality metric for the object with information about the local environment and the robot kinematics. For this solution to be effective, a database of off-line computed grasps is required, and the manipulator must be positioned such that the object is located well within the workspace of the pre-computed grasps. As the manipulator position is directly related to the degrees of freedom (DOF) of the base, the ability of the base to position the manipulator in a desirable location becomes crucial to successful pose selection. However, it is not always feasible to move the manipulator base due to environmental conditions and the limitations of base movement capabilities. One solution to this issue is to extend the workspace and the set of pre-computed grasps to encompass the object. It is not always feasible to extend the workspace due to the high dimensionality of the problem and the computational time required to compute poses for the extended grasp set.

There is a need for a more general solution for mobile manipulator pose selection that does not suffer from the limitations of inverse kinematics such as singularities and workspace reduction. There is also a need to have collision avoidance and joint limit checking integrated into the pose selection process to eliminate the redundancy caused by calculating poses followed by a validity checking process.

This paper presents an alternative to the inverse kinematics method of pose selection for mobile manipulators based on target tracking pose selection research [Webb, 2008] and autonomous exploring and mapping research [Paul *et al.*, 2009] for fixed manipulators. This optimsation pose selection method uses a series of task objectives that define the desired position of the end-effector in relation to the target object as well as collision avoidance and joint limit checking objectives. These objectives are combined into a cost function which is optimised to determine the grasp pose. Unlike inverse kinematics that requires convergence to an exact solution, the end-effector grasp pose is constrained to within an acceptable range meaning the chance of convergence is much higher. The result is an integrated and efficient pose selection and validity checking process that does not rely on any *a priori* knowledge of the environment and is not limited by a precomputed set of grasps or a workspace reduction.

The remainder of this paper is organised as follows: Section 2 provides an outline of the overall framework and background to the paper. Section 3 describes the steps needed to compute the grasp-scoring function. Section 4 demonstrates the improvement in performance of the optimisation pose selection method compared to inverse kinematics. Section 4 shows the results of testing in simulation and when combined with an existing Rapidly-exploring Random Tree (RRT) path planner and used on the RobotAssist 6 DOF manipulator.

2 Background

To successfully grasp an object, a manipulator pose, $\vec{Q}$, that corresponds to an adequate grasp primitive must be determined. The grasp primitive must specify the acceptable orientation range of the gripper and the acceptable distance limits to the object for successful end-effector pose selection. Using inverse kinematics to calculate $\vec{Q}$ is not always desirable as there is a possibility that the object is located on the edge of the manipulator workspace, and that there are obstacles present. Therefore, an optimising pose selection method can be used to find a pose which corresponds to a desirable grasp primitive. In order to perform pose selection, a relationship between the pose $\vec{Q}$ and the quality of the resulting grasp primitive is established. Through forward kinematics it is possible to compute the position and orientation of the end-effector. Based upon the manipulator model, the end-effector position and orientation is represented by the homogeneous transformation matrix, ${}^0T_f(\vec{Q})$, which is calculated by performing transformations through the $i \in \{1, \dots 6\}$ joint angles, q_i as shown below:

$$ {}^0T_f(\vec{Q}) = \prod_{i=1}^{6} {}^{i-1}T_i(q_i) \tag{1} $$

The matrix in Equation 2 represents the position and orientation of the end-effector in the robot base coordinate frame. This transform matrix, with end-effector position (in the Cartesian coordinate frame) $[x, y, z]^T$,

the end-effector pitch axis of rotation $\mathbf{N} = [n_x, n_y, n_z]^T$, yaw axis of rotation $\mathbf{O} = [o_x, o_y, o_z]^T$, and roll axis of rotation $\mathbf{A} = [a_x, a_y, a_z]^T$ and is visualised in Figure 1.

$$ {}^0T_f(\vec{Q}) = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

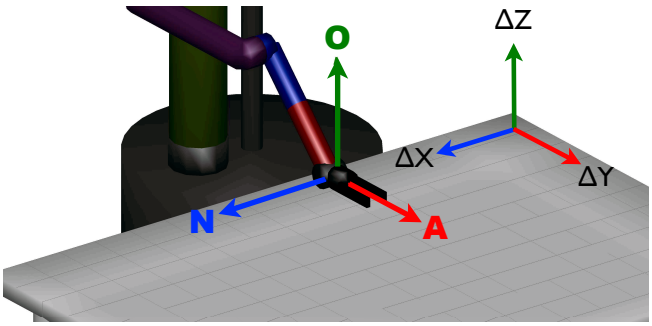

Figure 1: The end-effector position and **N**, **O**, **A** orientation components of the transform matrix.

The optimisation pose selection method is based on a fine tuning optimisation approach using the Levenberg-Marquardt's (LM) algorithm [Marquardt, 1963] instead of inverse kinematics. This algorithm interpolates between the Gauss Newton algorithm and the method of gradient descent, and is primarily used in the least squares curve fitting problem. To apply this optimisation technique to the presented method, each objective function will be formulated as a cost function, g_c, by incorporating pose objectives in sigmoid functions as shown in Figure 2. Each cost function will have bounding limits to constrain the final solution. A sigmoid function is appropriate since it enables both minimum and/or maximum objectives to be formulated. If, for a general constraint case, an angle θ is required to be less than a maximum angle θ_{max}, then this can be incorporated into a sigmoid function $g_c(\theta) = \frac{1}{1+e^{-(\theta-\theta_{max})}}$. If the solution is required to be greater than an angle θ_{min}, then $g_c(\theta) = 1 - \frac{1}{1+e^{-(\theta-\theta_{min})}}$. If the acceptable angular value is between two angles, θ_{min} and θ_{max}, then $g_c(\theta) = 1 - \frac{1}{1+e^{-(\theta-\theta_{min})}} + \frac{1}{1+e^{-(\theta-\theta_{max})}}$.

An objective function formulated as a sigmoid function is continuous with a range between 0 and 1. The objective function can be optimised (i.e. minimised) by using the LM algorithm. Due to the nature of sigmoid

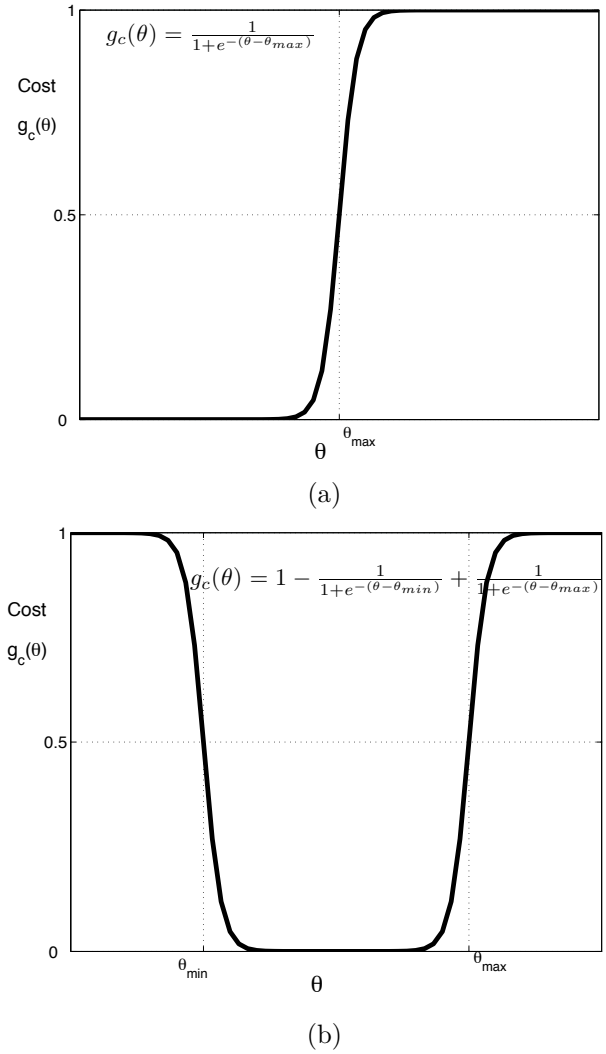

Figure 2: Sigmoid functions for a cost function, $g_c(\theta)$. (a) Cost function for a maximum constraint e.g. $\theta < \theta_{max}$. (b) Cost function for minimum and maximum constraints e.g. $\theta_{min} < \theta < \theta_{max}$.

functions, multiple objective functions can be combined by summing their squares. Therefore, the multiple objectives for a manipulator pose must be formulated, and then combined. The combined objective functions can then be optimised so as to determine a pose $\vec{Q}$, corresponding to an appropriate grasp primitive which will enable an object to be picked up.

3 Optimisation Pose Selection

3.1 Angular Pose Objectives

It is critical that the end effector is orientated correctly to ensure a successful grasp and manipulation of the target object. Incorrect orientation may result in the

gripper missing the object or being unable to grasp it. To ensure that the gripper orientation is acceptable, two angular objective functions are created to constrain the pitch and roll movement of the end effector along the normal vector and the approach vector respectively.

The roll constraint objective as shown in Figure 3(a) aims to minimise the angle ϕ (between the object normal vector and the normal vector of the end-effector) deviation from 90°. This objective function is described below:

$$\min_{\vec{Q}} g_1(\vec{Q}) = 1 - \frac{1}{1 + e^{-(\phi - \phi_{min})}} + \frac{1}{1 + e^{-(\phi - \phi_{max})}} \quad (3)$$

In this function, ϕ must lie between the upper (ϕ_{max}) and lower (ϕ_{min}) limits either side of 90°.

The pitch constraint objective as shown in Figure 3(b) aims to minimise the angle θ between the normal vector of the object and the orientation vector of the end-effector. θ_{max} represents the greatest acceptable angle between the end-effector orientation vector and the target object normal vector.

This objective function can be described by Equation 4:

$$\min_{\vec{Q}} g_2(\vec{Q}) = \frac{1}{1 + e^{-(\theta - \theta_{max})}} \quad (4)$$

3.2 Distance Pose Objectives

A critical task objective in the pose selection problem is the distance of the end effector to the target object. For the manipulator to achieve successful grasping of the target object, the grasp pose of the end-effector must be in close proximity so that the final approach movement is successful, as shown in Figure 3(c) and (d). To achieve successful grasping, the normal distance (Δd_n in Figure 3(d)), orientation distance (Δd_o in Figure 3(c)) and the approach distance (Δd_a in Figure 3(d)) need to be decoupled as separate objectives so that each can be optimised individually. This method yields a more optimised result than if they are combined in a 3D Euclidean distance measurement from end-effector to target object. The three objective functions for distance measurements are:

i. Distance along the end-effector normal vector, Δd_n:

$$\min_{\vec{Q}} g_3(\vec{Q}) = \frac{1}{1 + e^{-(d_n - dn_{max})}} \quad (5)$$

ii. Distance along the end-effector orientation vector, Δd_o:

$$\min_{\vec{Q}} g_4(\vec{Q}) = \frac{1}{1 + e^{-(d_o - do_{max})}} \quad (6)$$

iii. Distance along the end-effector approach vector, Δd_a:

$$\min_{\vec{Q}} g_5(\vec{Q}) = \frac{1}{1 + e^{-(d_a - da_{max})}} \quad (7)$$

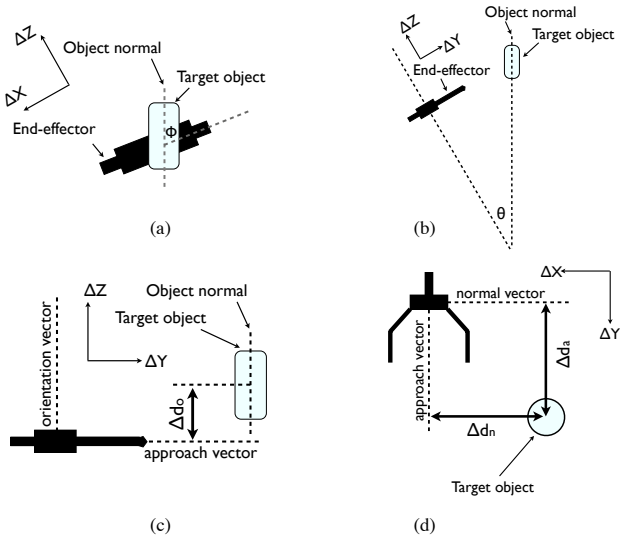

Figure 3: Task objectives for the cost function. (a) The end-effector roll objective function, $\phi_{min} < \phi < \phi_{max}$. ($b$) The end-effector pitch objective function, $\theta < \theta_{max}$. (c) The end-effector orientation distance objective function, $\Delta d_o < \Delta do_{max}$. ($d$) The end-effector normal distance objective function, $\Delta d_n < \Delta dn_{max}$ and the end-effector approach distance objective function, $\Delta da_{min} < \Delta d_a < \Delta da_{max}$.

3.3 Manipulator Joint Limitations

The manipulator joint configuration (i.e. manipulator pose) is described by the vector of joints, $\vec{Q}$, and must fall within the physical angular limitations. For each of the $i \in \{1, \ldots 6\}$ joints, these are defined as the positive maximums, $q_{i,max}$, and negative minimums, $q_{i,min}$. For this pose selection, physical joint limitations must be parameterised so the grasp primitive solution is valid. The joint limits are enforced in implementation to ensure compliance. Since the LM algorithm solves the cost functions by minimising the sum of the squares, this objective function, $g_6(\vec{Q})$, has combined the six pairs of joint constraints into one function by summing the squares as follows:

$$\min_{\vec{Q}} g_6(\vec{Q}) = \sum_{i=1}^{6} \left(1 - \frac{1}{1+e^{-(q_i - q_{i,min})}} + \frac{1}{1+e^{-(q_i - q_{i,max})}}\right)^2 \quad (8)$$

3.4 Safe Manipulator Pose Selection

Safe manipulator movement is another vital consideration during pose selection. Safe movements are those that can avoid any intersection (collision) of the manipulator with obstacles. In order to improve the efficiency of collision-free pose selection, ellipsoidal virtual bounding fields [Chotiprayanakul *et al.*, 2007] can be used to perform safety checks. The ellipsoidal field of the ith manipulator link has the centre vector $\mathbf{p}_{c,i}$, a vector of equatorial radii, $[a_{e,i}, b_{e,i}]$, and a polar radius, $c_{e,i}$.

The position and orientation of an ellipsoid in the global coordinate frame, which represents a link, i, of the manipulator is specified by the 4×4 rotation and translation homogeneous transformation matrix, ${}^0T_i(\vec{Q})$, calculated as:

$${}^0T_i(\vec{Q}) = \prod_{j=1}^{i} {}^{j-1}T_j(q_j) \quad (9)$$

Although the transformation for the ith ellipsoid is presented as a function of the pose, $\vec{Q}$, only the first i joints are used when transforming the ith coordinate frame. Each obstacle is represented by a position vector, $\mathbf{p} \in \mathsf{P}$, where P is a set of points (positions) in the environment. $\mathbf{p}$ can be transformed into the ith ellipsoid's coordinate frame (where the ellipsoid is at the origin and an obstacle is positioned and orientated relatively) by using the inverse of the transformation matrix. A transformed point is thus denoted as $\mathbf{p}^{\,{}^0T_i(\vec{Q})^{-1}}$. The algebraic distance can then be calculated from the ellipsoid to each point. The minimum algebraic distance, $dist(\vec{Q})$, is a function of $\vec{Q}$ for points $\mathbf{p} \in \mathsf{P}$, and all the encompassing ellipsoids of the manipulator as shown in Equation 10.

The objective function, $g_7(\vec{Q})$, calculates a grasp primitive, $\vec{Q}$, which maximises the smallest algebraic distance ($dist(\vec{Q})$) from all a manipulator's ellipsoids to all obstructing points. If a bounding ellipsoidal field around the manipulator at a grasp pose had its smallest algebraic distance less than unity ($g_7(\vec{Q}) \leq 1$), it is unacceptable since this means a collision will occur. This objective function ensures that safe poses and motions are considered desirable, and constrains the grasp solution to safe poses (i.e. $g_7(\vec{Q}) > 1$). The collision checking sigmoid cost function is:

$$\min_{\vec{Q}} g_7(\vec{Q}) = 1 - \frac{1}{1 + e^{-\left(dist(\vec{Q})-1\right)}} \quad (11)$$

The seven objective functions presented have each been formulated for different purposes. A proposed grasp primitive which minimises the approach angle of the end-effector may be outside the workspace of the manipulator. Alternatively, a collision free grasp primitive may not be close enough to the object. It is therefore reasonable to assume that objectives may be in conflict. An optimisation approach is required to combine the cost functions so that a grasp primitive can be selected. The combined cost function is thus defined as:

$$\min_{\vec{Q}} \mathbf{g}(\vec{Q}) = \sum_{i=1}^{7} \left(g_i(\vec{Q}) - \hat{g}_i\right)^2 \quad (12)$$

where: the desired value, $\hat{g}_i$, for all $i \in \{1, \ldots 7\}$ cost functions is zero (i.e. $\hat{g}_i = 0$). The pose selection algorithm is designed for the rapid discovery of a valid joint configuration for a grasp pose. LM is an established iterative technique which can be used to locate the minimum of the sum of squares of a non-linear function, $\mathbf{g}(\vec{Q})$, as shown in [Kelley, 1999]. The LM algorithm is used to determine the step direction towards $\vec{Q}$, using the Jacobian of the vector of cost function, $\mathbf{g}(\vec{Q})$, and to return the new joint angles.

4 Results

The optimisation pose selection method was used to solve a number of pose selection problems of varying difficulty. Experiments were conducted using a simulation model of a 6DOF Exact Dynamics iArm developed using Denavit-Hartenberg parameters. As well as this, a test was conducted on the actual 6DOF Exact Dynamics manipulator in a real world environment.

4.1 Simulation Experiment

To adequately test the optimisation pose selection method it is necessary to conduct grasp scenarios with the object placed across a large range of the manipulator workspace in the presence of obstacles. An experiment was set up to place an object in 78 arbitrary positions on a table within the workspace of the manipulator. By doing this, a large selection of reachable and non-reachable positions could be set as shown in Figure 4. The obstacle location was chosen to be in an area which causes significant interference for the manipulator. The optimisation pose selection method does not rely on *a-priori* knowledge of the obstacle position, meaning the obstacle can easily be moved to other locations in the workspace. This experiment was designed to

$$dist(\vec{Q}) = \min_{\mathbf{p}\in\mathsf{P}} \left(\min_{i\in\{1,\ldots 6\}} \left(\left({}^{0}T_i(\vec{Q})^{-1} - \mathbf{p}_{c,i} \right)^T \cdot \begin{pmatrix} a_{e,i}^{-2} & 0 & 0 \\ 0 & b_{e,i}^{-2} & 0 \\ 0 & 0 & c_{e,i}^{-2} \end{pmatrix} \cdot \left({}^{0}T_i(\vec{Q})^{-1} - \mathbf{p}_{c,i} \right) \right) \right) \tag{10}$$

test whether the optimisation pose selection method was effective in overcoming singularity limitations and obstacle avoidance.

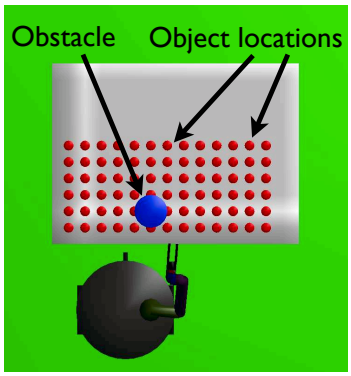

Figure 4: The locations of the 78 target object positions and the obstacle in the manipulator workspace.

An inverse kinematics solution to pose selection based on the "ikine" function in Matlab was used as a control test for the first part of this experiment. A database of 780 possible inverse kinematic frames were precomputed, allowing 10 frames per discrete object position as shown in Figure 5. If the workspace was reduced or the computational time was increased, more frames would have to be calculated to increase the probability of finding a solution. However, for this experiment a large workspace was utilised and a similar computational time for both the optimisation pose selection and inverse kinematics was maintained. After inverse kinematics determined the successful poses from the precomputed frame set, collision checking and joint limit checks were conducted to eliminate the invalid poses from the results. The inverse kinematics solution returned 578 pose configurations from the 780 frames and after collision checking and joint limit checks only 140 of the 780 frames were found to be valid. To be considered a successful grasp, at least one frame from the frame set needed to return valid for an object in a certain location. Figure 6 shows the number of frames that returned valid for the different object positions. The results from this graph indicate that there is a large probability that the inverse kinematics solution will not return any valid frames due to the presence of obstacles and the target object positioned near the edge of the manipulator workspace. The graph in Figure 6 also shows that there were no instances where the inverse kinematic method return all ten frames as valid. This confirms the necessity to produce an offline database of grasp sets for the inverse kinematics method to increase the probability of finding a valid grasp pose. For the second part of this experiment, the optimisation pose selection method was used to determine the quantity of valid poses for the same 78 object locations. The results from the optimisation pose selection test were compared to the results of the inverse kinematics test for the 78 object positions as shown in Table 1.

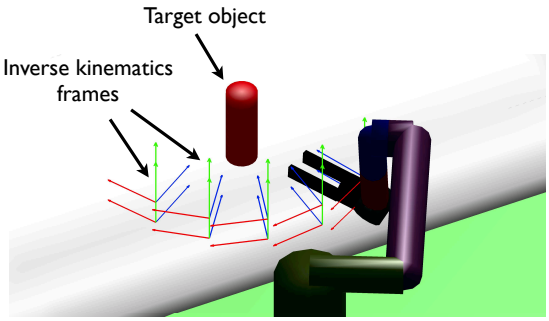

Figure 5: The ten inverse kinematics precomputed frames used for each target object position.

The results in Table 1 show that there is a 34% improvement in the ability of the end-effector to reach a successful grasp pose within the manipulator workspace when using the optimisation pose selection method compared to the inverse kinematics method. The probability of finding a solution using the inverse kinematics method could be increased using a larger set of precomputed grasps, however workspace size and computational efficiency would be sacrificed as previously discussed.

4.2 Real World Experiment

The optimisation pose selection method was combined with an existing RRT path planner and tested on the RobotAssist platform used in the 2010 RoboCup@Home competition with an Exact Dynamics 6DOF manipulator. For this experiment, a series of object locations were

	Inverse kinematics control test	Optimisation pose selection method
Number of valid poses calculated	35	62
Total number of object positions	78	78
Percentage of poses found in test	45%	79%

Table 1: The results of the optimisation pose selection method and inverse kinematics method comparative test. The table shows the number of valid poses found from the 78 target object positions for both methods.

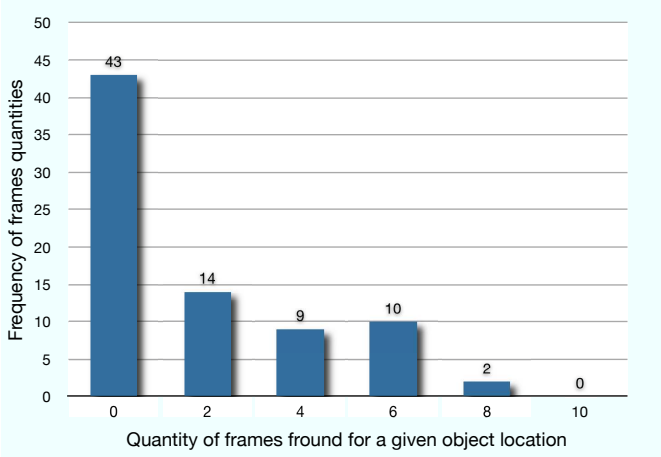

Figure 6: The number of valid frames calculated using inverse kinematics for each target object position.

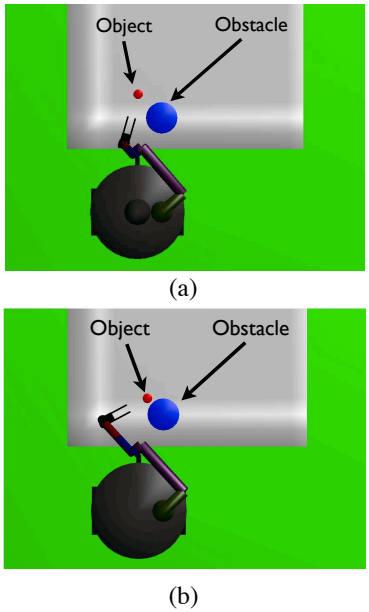

Figure 7: (a) & (b) show the simulated results of the optimisation pose selection method successfully finding a valid pose for an object behind an obstacle in the workspace.

selected that are difficult for pose selection due to the presence of obstacles in the workspace of the manipulator. Valid poses were calculated using the optimisation pose selection method for each object position in simulation and on the Exact Dynamics 6DOF manipulator. Figure 7 (a) and (b) show successful grasp poses being calculated using the optimisation pose selection method for objects positioned behind an obstacle. Figure 8(a) shows the optimisation pose selection method successfully being used on the Exact Dynamics 6DOF manipulator to find a valid grasp pose for objects positioned behind an obstacle. Figure 8(b) shows the RobotAssist platform successfully grasping a target object as a result of the optimisation pose selection method calculating a valid grasp pose.

5 Conclusion

This paper has presented an alternative to the inverse kinematics method of pose selection for mobile manipulators. By forming end-effector, collision avoidance and joint limit checking task objectives into a cost function using sigmoid functions, an optimised grasp pose can be calculated. This optimisation pose selection method is not affected by singularities and can combine collision checking and joint limit checks into the grasp selection process. The presented method can quickly and efficiently find valid grasp poses within a manipulator workspace without the need for an offline database of precomputed grasps. The effectiveness of the presented method has been demonstrated by conducting a pose selection experiment both in simulation and in a real world environment for target objects located throughout the workspace of the manipulator.

Future work will focus on improving the cost function to incorporate self collision checking and extend it to be capable of picking up non-uniform objects such as boxes and wine glasses. Future work will also focus on considering and measuring the computational efficiency improvements that result from using the optimisation pose selection method of grasp identification compared to inverse kinematics.

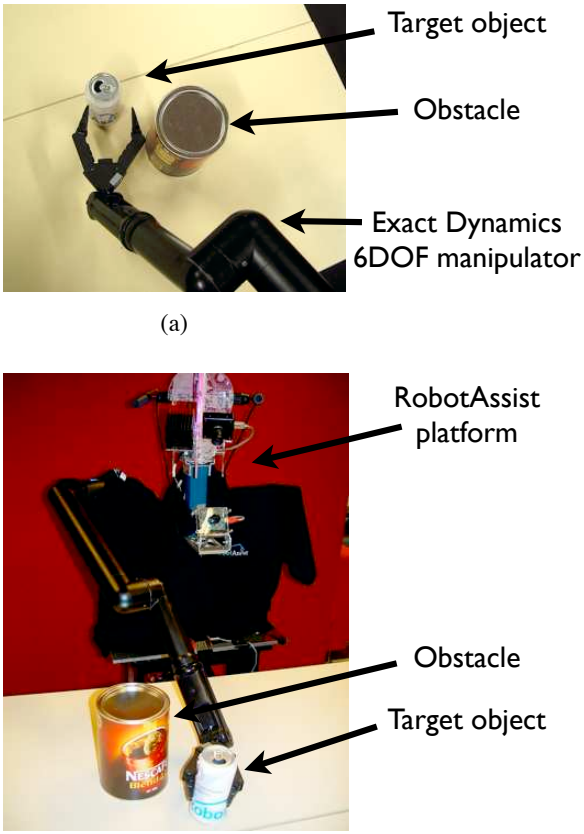

Figure 8: The RobotAssist platform successfully grasping an object behind an obstacle in a real world environment.

Acknowledgements

This work is supported by the ARC Centre of Excellence for Autonomous Systems, RobotAssist and the University of Technology, Sydney.

References

[Berenson *et al.*, 2007] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *7th IEEE-RAS International Conference on Humanoid Robots*, pages 42 –48, 2007.

[Berenson *et al.*, 2008] D. Berenson, J. Kuffner, and H. Choset. An optimization approach to planning for mobile manipulation. In *ICRA 2008. IEEE International Conference on Robotics and Automation*, pages 1187 –1192, 2008.

[Bertram *et al.*, 2006] D. Bertram, J. Kuffner, R. Dillmann, and T. Asfour. An integrated approach to inverse kinematics and path planning for redundant manipulators. In *ICRA 2006. Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 1874 –1879, 2006.

[Bicchi and Kumar, 2000] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *ICRA 2000. IEEE International Conference on Robotics and Automation*, volume 1, pages 348 –353, 2000.

[Cheng *et al.*, 1995] F. Cheng, T. Hour, T. Chen, Y. Sun, and F. Kung. Study and resolution of singularities for a 6-DOF PUMA manipulator. In *IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 5, pages 4410 –4415, 1995.

[Chotiprayanakul *et al.*, 2007] P. Chotiprayanakul, D. K. Liu, D. Wang, and G. Dissanayake. A 3-Dimensional force field method for robot collision avoidance in bridge maintenance. In *Proc. 24th Intl. Symp. on Automation and Robotics in Construction*, pages 139–145, 2007.

[Dinesh K. Pai, 1992] M. C. Leu Dinesh K. Pai, Member. Genericity and singularities of robot manipulators. *IEEE Transactions on Robotics and Automation*, 8(5), 1992.

[D'Souza *et al.*, 2001] A. D'Souza, S. Vijayakumar, and S. Schaal. Learning inverse kinematics. In *Proceedings. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 298 –303, 2001.

[Hsiao and Lozano-Perez, 2006] K. Hsiao and T. Lozano-Perez. Imitation learning of whole-body grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5657 –5662, 2006.

[Kelley, 1999] C. T. Kelley. Iterative methods for optimization. In *SIAM Frontiers in Applied Mathematics*, Philadelphia, 1999. SIAM Press.

[Klein and Huang, 1983] C. Klein and C. Huang. Review on pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on System, Man and Cybernetics*, 13(3):245–250, 1983.

[Kuffner and LaValle, 2000] Jr. Kuffner, J.J. and S.M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proceedings. ICRA '00. IEEE International Conference on Robotics and Automation*, volume 2, pages 995 –1001, 2000.

[Marquardt, 1963] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[Mason, 2001] M.T. Mason. *Mechanics of Robotic Manipulation*. Cambridge, MA:MIT Press, 2001.

[Miller *et al.*, 2003] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation*, volume 2, pages 1824 – 1829, 2003.

[Morales *et al.*, 2006] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5663 –5668, 2006.

[Paul *et al.*, 2009] G. Paul, N. Kirchner, D. K. Liu, and G. Dissanayake. An effective exploration approach to simultaneous mapping and surface material-type identification of complex 3d environments. *Journal of Field Robotics, Special Issue on Three-Dimensional Mapping*, 26(11-12 SI):915–933, 2009.

[Pelossof *et al.*, 2004] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An SVM learning approach to robotic grasping. In *Proceedings. ICRA '04. 2004 IEEE International Conference on Robotics and Automation*, volume 4, pages 3512 – 3518, 2004.

[Shin-ichi and Satoshi, 2000] T. Shin-ichi and M. Satoshi. Living and working with robots. *Nipponia*, 13:7–12, 2000.

[Webb, 2008] S.S. Webb. *Belief Driven Autonomous Manipulator Pose Selection for Less Con- Belief Driven Autonomous Manipulator Pose Selection for Less Controlled Environments*. PhD thesis, University of New South Wales, Australia, 2008.

[Yakey *et al.*, 2001] J.H. Yakey, S.M. LaValle, and L.E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *Robotics and Automation, IEEE Transactions on*, 17(6):951 –958, 2001.