

Path Optimization for Abstractly Represented Tasks with Respect to Efficient Control

Susanne Petsch and Darius Burschka

Abstract—In order to be able to replace a human operator, a robotic manipulation system needs to deal with a variety of possible actions. These actions may be more or less constrained in their motion profile and in the accuracy of the transport goals. The robotic system can make use of some of this variation to simplify the control to improve the efficiency of the generated motion. Nevertheless, the human’s intention behind the manipulation may not change. We introduce the *Elastic Power Path* to optimize paths with respect to efficient control in the context of abstractly represented tasks.

Our experiments show, that the proposed Elastic Power Path is an efficient method to achieve this aim. The magnitude and the number of turnarounds of the accelerations along the path are significantly reduced.

I. MOTIVATION

A human, who repeats an action, does usually not use the same paths all the time, but a variety of paths. However, the human has a certain intention regarding the action. This intention is not changing during the repetitions. It can be encapsulated in the characteristic properties of an abstract task representation. When a robotic system is supposed to perform a task instead of the human, it needs to consider the characteristic properties of the task. At the same time, the system can make use of the freedom in planning, since it is not restricted to a certain path. A variety of paths fulfilling the characteristic properties are possible. Of course, it is desirable to select the most efficient one to enable an easy and efficient control. Furthermore, energy can be saved and the strain on the hardware can be reduced. Hence, each joint of the robot should move slowly and smoothly. Abrupt turnarounds of the acceleration should be avoided. But how should the path look like to support efficient dynamics?

To sum up, we want to make use of the freedom in planning which comes along with the abstract representation of tasks. We want to optimize the path with respect to efficiency in control.

We introduce the *Elastic Power Path* for this purpose. In analogy to an elastic band, the Elastic Power Path has a certain elasticity. When an elastic band is stretched, additional energy is required. In the context of efficiency, we want to minimize this additional energy. The less stretched the band, the closer desired optimum is. This depends, of course, on the elasticity of the band. Analogously to an elastic band, the Elastic Power Path can be stretched and relaxed. Its elasticity

This work was supported by a cooperation with the German Aerospace Center (DLR).

Susanne Petsch and Darius Burschka are with the Machine Vision and Perception Group, Department of Informatics, Technische Universität München, 85748 Garching, Germany
{petsch,burschka}@in.tum.de

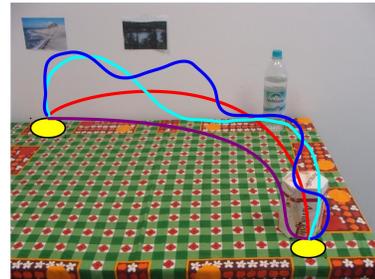


Fig. 1. Different path configurations are shown for a manipulation between two areas (yellow). Which path is the most efficient one with respect to control?

depends on the dynamics of the robot along the path. The more efficient the dynamics, the closer the Elastic Power Path is to its optimum. Our aim is an optimal path with respect to efficient dynamics. We measure the required power in the style of a hiker climbing up and down hills. The required power is increasing, if more hills have to be climbed or if higher hills have to be climbed. The speed and acceleration profiles play an important role regarding the power. In the context of path optimization for a manipulator, we focus on the speed and acceleration of the joints. The less often the direction of the acceleration is reversed and the smaller its magnitude, the less power is needed.

It is important to point out, that we do not analyze all possible configurations of the robot in the entire workspace. The concept of the Elastic Power Path allows an efficient optimization for the desired task: Just promising configurations of the robot are considered during the optimization. Moreover, the optimization does not depend on a specific manipulator structure. It is not necessary to find appropriate joints in the manipulator structure for each (sub-)motion.

We presented already an analysis tool, which uses the abstraction of the human actions to generate paths with efficient motion profiles [1]. The comparison of a robot’s dynamics for different paths showed, that certain path properties are preferred to support efficient control. Furthermore, we observed, that the intuitive solution did not necessarily agree with the results optimizing for efficiency. This shows, that it is worth to investigate in a concept for path optimization with respect to efficient control.

We build on the concept of Functionality Maps [2] which represent observed manipulation tasks in an abstract manner. Different properties of manipulation tasks are stored there, as, e.g., the type of manipulation or characteristic areas (Location Areas), which are start or end points of manipulations.

The paper is structured as follows: After an overview of the related work, the approach is presented. In the approach, the elasticity of the path is explained first. Afterward, the Elastic Power Path is introduced. The description of the optimization of the path is the last part of the approach. The data, the implementation and the results of our experiments are described thereafter. We end with conclusions.

II. RELATED WORK

It is important to point out, that we do not aim to search a path (e.g., [3], [4]). We want to optimize a path of an *abstractly represented task* with respect to *efficient* control.

Path optimization has been done with respect to different criteria. For example, points on the path were optimized to increase the distance to obstacles [5]. Kinematics singularities were considered in [6]. Smith *et al.* proposed an optimal path planning for surveillance with temporal-logic constraints [7]. Other authors include dynamics in the optimization process. In [8], the authors dealt with the complicated dynamics of large space manipulators. Rieswijk *et al.* incorporated actuator and jerk constraints in [9].

If we want to optimize the path with respect to efficient control, a path optimization with respect to a minimal distance between the start and end points in 3D space could be an intuitive solution. As we have shown in previous experiments [1], the shortest path is not necessarily the most efficient one with respect to efficient dynamics. Hence, we propose the Elastic Power Path to optimize the path configuration with respect to efficient control.

The name of our concept “Elastic Power Path” might cause an association with the “Elastic Strip Framework” [10], especially in the context of path planning. Both concepts have the elasticity of the path in common. However, the aims are different. We want to use the elasticity of the path to improve the efficiency in control. In [11], [12], a minimization of energy (the instantaneous kinetic energy of the robot) is processed among others. A special Jacobian is used there. In contrast, our optimization is independent of any Jacobian J . Moreover, δx and $\delta \theta$ are unknown (e.g., in $\delta x = J(\theta)\delta \theta$). Furthermore, we use a global optimization method to overcome local minima. The aim of our path optimization is efficient control under the consideration of the abstract characteristic properties of a desired manipulation.

III. APPROACH

We want to optimize a path for a manipulator with respect to efficient dynamics. A clear definition about the used variables and definitions is important. Therefore, we give a short overview of them.

We describe the robotic system in the DH-convention suggested by Denavit and Hartenberg [13] in the form shown in [14]. The manipulator system consists of D joints.

Each joint d contributes a certain linear velocity vector v_d to the overall velocity v of the end-effector:

$$v = \sum_d v_d \quad (1)$$

The linear velocity v_d of a rotational joint d can be computed by

$$v_d = \omega_d \times (P_{ee} - P_d) \quad (2)$$

with P_{ee} as the position of the end-effector, P_d as the position of joint d and ω_d as the rotational velocity caused by joint d :

$$\omega_d = \dot{\theta}_d \cdot \hat{Z}_d \quad (3)$$

with \hat{Z}_d as 3-dimensional unit vector of the z-axis of joint d and $\dot{\theta}_d$ as magnitude of the angular rate (see, e.g., [14]).

The acceleration is labeled with Δv_d , indicating its meaning as difference between two consecutive speed values.

A. Elasticity of the Path

Our basic idea can be illustrated through an elastic band with linear elasticity. The elasticity can be described through Hooke’s law, similarly to a linear spring:

$$F = -k \cdot x \quad (4)$$

with x as the displacement from the equilibrium position, k as the spring constant and F as the resulting force.

We use the elasticity of the path as illustration of our idea, therefore, we assume that the displacement is within the elastic range. In context of the elasticity of the path, the spring constant k depends on the efficiency of the system’s dynamics. The displacement x describes the distance between the current configuration of the path points and their ideal configuration.

Hook’s law of elasticity can be seen as optimization function, which optimizes the system’s configuration with respect to energy. In contrast to a real elastic band, the optimization function of the Elastic Power Path is much more complex. First, the elasticity depends on the efficiency of the dynamics. Many local minima can occur in the optimization function. Second, the ideal configuration of the path points, resp., the displacement x is unknown. Therefore, the force F cannot be computed directly. This results in a chicken-and-egg problem: The displacement x is unknown, consequently, the force F cannot be computed and vice versa.

B. The Elastic Power Path

Similarly to Hook’s law, we want to optimize the path configuration of a manipulator system with respect to energy. In our context, the energy is consumed by the manipulator to perform the desired task. As motivated at the beginning of the paper, we want to minimize the required energy.

The manipulator spends the energy to move its joints. Hence, the movement of the joints should be performed at smooth and low speed. This requires, in turn, a smooth acceleration profile. The acceleration should not only be smooth, but it should also have a low magnitude. Moreover, zero crossings should be avoided, since a change of the acceleration sign refers to a large energy loss. The movement in the previous direction has to be decreased or even stopped (braking), while a movement into the other direction has to be built up. Moreover, braking means that energy is necessary to stop the movement in the previous direction, which resulted absurdly from an earlier investment of energy.

We model the required energy in the style of a hiker climbing up and down hills. Hence, we compute the power the hiker would need to climb up and down all desired hills. In general, the power P is computed through

$$P = \frac{\Delta E}{\Delta t} \quad (5)$$

with time t and the kinetic Energy E as

$$E = \frac{1}{2}mv^2. \quad (6)$$

Using the derivative of the kinetic Energy E , the power P can be computed by

$$P = \frac{\Delta E}{\Delta t} = m \cdot \frac{v \cdot \Delta v + 0.5 \cdot (\Delta v)^2}{\Delta t}. \quad (7)$$

In our case, the relevant acceleration is $\Delta v_{d,i}$, which is the maximal acceleration contributed by joint d within Δt_i . Δt_i is the required time to climb up and down hill i . The hiker starts and ends his/ her tours at consecutive zero points in the acceleration profile (p_i and p_{i+1}). Each $\Delta v_{d,i}$ reflects, then, one peak in the acceleration curve. If the acceleration curve proceeds below zero-level, it is mirrored on the zero-level to ensure that all peaks are counted equally later. It is important to point out, that the zero points do not change. Fig. 2 illustrates this procedure.

The power $P_{d,i}$ can be computed for each hill i in the acceleration profile of joint d . However, the relevant part of $P_{d,i}$ is the fraction in our case, since m is always constant. Hence, just a change of a variable in the fraction enables a change of the magnitude of P . m just scales the result. Consequently, we want to minimize the following Objective Function O_d for each joint d along all points p_i on the path:

$$O_d = \sum_i \frac{2 \cdot |v_{d,i}| \cdot |\Delta v_{d,i}| + (\Delta v_{d,i})^2}{\Delta t_i}. \quad (8)$$

We use absolute values due to the described mirroring of $\Delta v_{d,i}$ on the zero-level to create the hills. The absolute values of $v_{d,i}$ are used to enable an equal counting of all peaks. The original fraction has been multiplied with a factor of two for esthetic reasons.

The overall Objective Function O is the sum of the Objective Functions O_d for each joint d :

$$O = \sum_d O_d = \sum_d \sum_i \frac{2 \cdot |v_{d,i}| \cdot |\Delta v_{d,i}| + (\Delta v_{d,i})^2}{\Delta t_i}. \quad (9)$$

C. Optimization of the Path

Our aim is to optimize the path in the context of an abstractly represented task. As already mentioned in the Motivation, we want to build on the concept of the Functionality Map [2]. This Map contains, e.g., the characteristic places of manipulations (Location Areas) and information about the type of the manipulation. Here, we make use of the Location Areas and the distinction between a manipulation of a pushed object on the table and a manipulation of a lifted object. These characteristic properties have to be preserved during

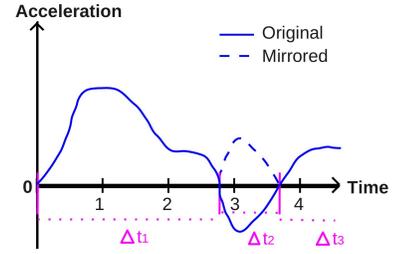


Fig. 2. The figure shows the original acceleration profile in blue. The negative values in the curve are mirrored at zero-level (dashed blue curve). Each zero point (vertical magenta lines) forms a border of a Δt_i . The black numbers are the original time steps. They refer to the time step x , when the corresponding point p_x is reached along the path. It is clearly visible, that the original time steps do not necessarily form the borders of Δt_i in the Objective Function O (Eq. 9).

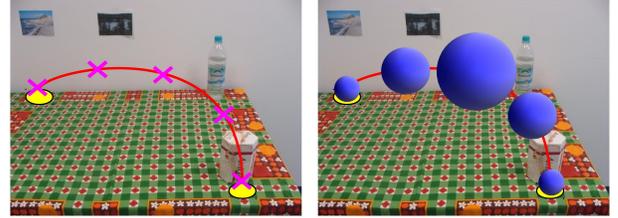


Fig. 3. The original points of the path are depicted on the *left* (magenta crosses). The corresponding spheres are visualized on the *right* in blue. Each point is just allowed to move within its corresponding sphere.

the optimization. For example, the path of a pushed object should be a path of a pushed object after the optimization of the path, too. Similarly, the path of the lifted object should still refer to a path of a lifted object.

Hence, we limit the possible configuration of the path. Of course, the start and end point of a path are hardly allowed to change. Moreover, points close to the start, resp., end point should lead to the start, resp., end point. The closer a point to the start or end point, the less freedom should be available for the path configuration. Therefore, we introduce a sphere S_x around each point p_x of the path. The sphere comprises the area, within which the point p_x is allowed to be placed during the optimization. The radius r_x of the sphere S_x depends on the position of p_x within the path. The further away p_x from the start and end point, the larger r_x .

If the original points p_x are uniformly distributed along the path, the radius r_x can be computed as follows:

$$r_x = \begin{cases} C \cdot x & \text{if } x \leq \frac{X}{2}, \\ C \cdot (X - x + 1) & \text{otherwise.} \end{cases} \quad (10)$$

C is a constant basic distance, X refers to the overall number of points along the path and x is the index of p_x . The index x is increasing along the path and it starts with $x = 1$ at the start point. Fig. 3 illustrates the spheres along a path. The initial points on the half circle are just chosen to ensure a path for a lifted object. The final points p_x do not need to stay on a circle, since they can be placed arbitrarily within each sphere S_x . In the case of a pushed object, we have to ensure, that each point stays at the original height above the plane on which the object is pushed. Therefore, the sphere S_x

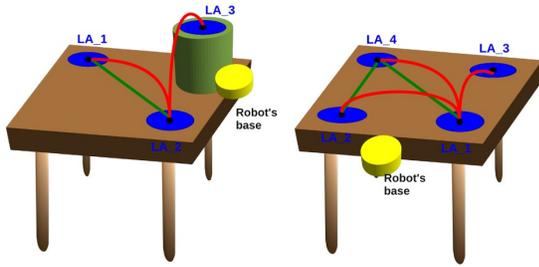


Fig. 4. Data set I (left) and II (right): The lines/ curves between the blue Location Areas refer to trajectories of lifted objects (red) and trajectories of pushed objects (green). The robot's base positions are symbolized in yellow.

is reduced to a circle at the desired height above the plane.

IV. EXPERIMENTS

Our experiments are based on a object manipulation scenario. The used data consists of two data sets, which were extracted in [2]. Both data sets are illustrated in Fig. 4.

A. Implementation

The implementation is done in C/C++. The stochastic approach for global minimization was presented in [15]. We use the implementation by Oliver Ruepp [16].

The manipulator consists of three rotational joints perpendicular to each other ($D = 3$). All links have a length of 300 mm (DH-Parameter: d_1, d_2, a_3)

The possible change of a position on the path is limited to the corresponding sphere S_x . We allow each joint d to change its position partially within the sphere. Hence, the contributed part of a joint is limited to r_x/D . The Jacobian is used to transfer the allowed part r_x/D to the configuration space. This transferred allowed part gives us the new range, within which the joints can be moved. The optimization is processed within these new ranges. The constant basic distance is chosen as $C = D \cdot 10.0$ mm. This choice limits the change of the start and end points, while allowing a large search space for the points in the middle of the path. The path consists of $X = 5$ points. We choose a relatively small number of points, since we want to optimize the overall path in a very short computation time. Intermediate points can be computed after the optimization by a simple partial movement of the joints if desired.

In order to keep the paths of pushed objects on the original height h_o above the plane, we add a residual R_x for each point p_x to the Objective Function O :

$$R_x = \begin{cases} 0 & \text{if } \|h_o - h_x\| \leq t_h \vee \\ A \cdot \|h_o - h_x\| & \text{otherwise.} \end{cases} \quad (11)$$

h_x is the current height of the point p_x above the plane. We introduce a tolerance t_h around the desired height h_o . The placement of points outside the region of the tolerance is punished with a factor of $A = 10.0$. We use the normal vector of the corresponding table plane to compute the desired and current height of the points. Of course, $R_x = 0$ for objects which are not pushed during the manipulation.

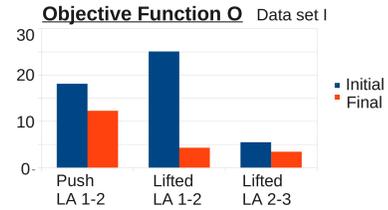


Fig. 5. Result of the Objective Function, data set I. The figure shows clearly, that the values of the Objective Function decrease from their blue initial values to the orange final values after the optimization (push/ lifted = pushed/ lifted object along the path, LA = Location Area).

The estimation of the inverse kinematics is also done through the stochastic approach for global minimization in [15]. The original path points are uniformly distributed along a line (if the object is pushed) or along a half circle which is positioned upright above the table (if the object is lifted). We estimate the three best configurations for the start point to position the manipulator as close as possible to the desired start point. The path optimization is processed for all three configurations. The best one is taken at the end.

Just the grasp has to be extremely precise (tolerance below 5 mm) in our application. Once the object has been grasped, we can extend the tolerance. The points after the start point are considered as reached by a certain configuration, if the distance of the real end-effector position to its desired position is smaller than a tolerance t . The points of data set I are reached with a tolerance t of 50.0 mm. The tolerance had to be extended to 75.0 mm for the data set II, to ensure, that all end-effector positions are within the tolerance.

In our experiments, we want to compare the dynamics of the robot along the original path and along the optimized path. In order to show the performance of our system, we need to determine configurations at the original points, which should already be advantageous with respect to dynamics and not arbitrarily. Hence, we determine just an arbitrary start configuration of the manipulator and search for consecutive configurations which are close to their corresponding ancestors along the path.

The tolerance t_h for the desired height above the table is set to 5.0 mm. Similarly to the tolerance t , tolerance t_h had to be extended to 25.0 mm for data set II, to ensure, that all end-effector positions are within the tolerance.

B. Results

The overall results of the optimization are depicted in Fig. 5 and 6. They show clearly, that a modification of the path configuration leads a significant improvement of the efficiency in dynamics in nearly all cases.

More details of the results are shown in Fig. 7-10. They show the change of the acceleration peaks for each single joint along the path. A look at the change of the acceleration peaks is most of the times enough to see the improvement after the optimization clearly (see, e.g., Fig. 7).

Just the optimization of the path of the pushed object between Location Area (LA) 1 and 2 requires an additional look at the change of the speed to see the improvement

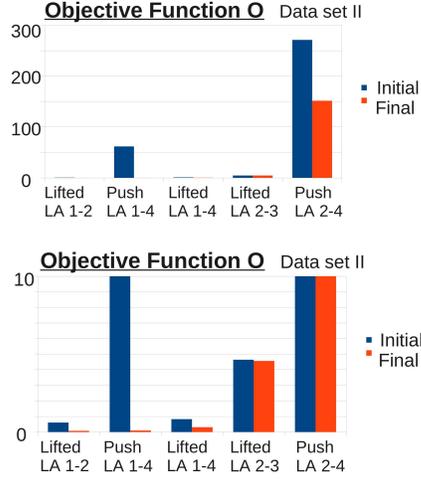


Fig. 6. Result of the Objective Function, data set II. The figure on the bottom shows the same results as the figure on the top. However, the figure on the bottom has a smaller data range on the vertical axis for a better illustration of the smaller values. As is can be seen, most of the values can be significantly reduced. This shows, that the modification of most of the paths has a strong effect. Just one path (lifted object between LA 2 and 3) can hardly be improved with respect to dynamics.

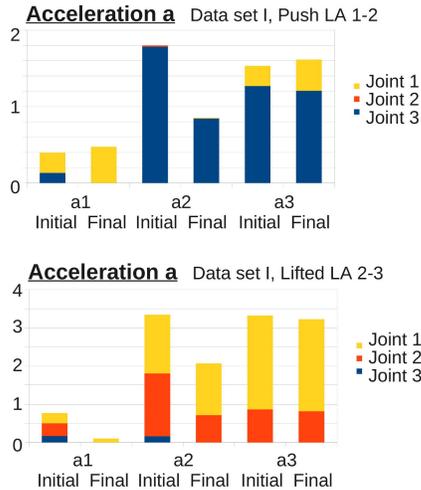


Fig. 7. Acceleration values per joint and per acceleration time step a_x along the path (in rad/s^2). The acceleration time step a_x refers to the time between point p_x and point p_{x+2} . For each acceleration time step a_x , the acceleration values at the beginning and the end of the optimization are shown (initial, resp., final values). The depicted acceleration values are the $\Delta v_{d,i}$ values which are relevant for the final Objective Function O (Eq. 9) at their time of occurrence x . Hence, the peaks of the hills in the acceleration profile are shown. The acceleration peaks of the path for the pushed object between Location Area (LA) 1 and 2 show a significant reduction of the acceleration of the third joint at a_2 (top figure). Moreover, the blue peak of the acceleration at a_1 is not existing any more after the optimization. This means, that the original hill is merged with another hill, in this case the hill with the the peak at a_2 . If hills have been merged, an undesired zero crossing has disappeared. Such a merge of hills is one of the desired aims of our approach. Some of the depicted acceleration values are very small, so that they are hardly visible (e.g., joint 2 on the top). Very desirable results are shown on the bottom: Most of the peaks are clearly reduced.

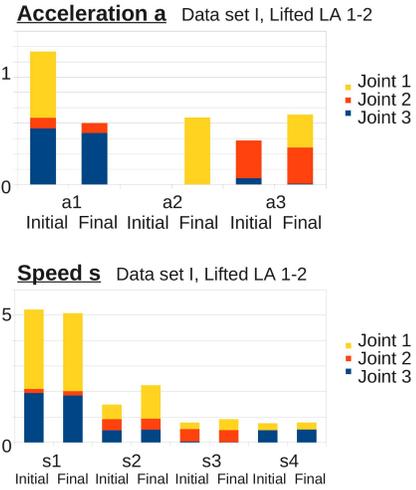


Fig. 8. Acceleration values per joint and per acceleration time step a_x (top), similarly to Fig. 7 and the corresponding speed values (bottom). The improvement of the peaks of the acceleration becomes just clear in the comprehensive survey with the speed values. Instead of an acceleration peak of the first joint at the beginning at a_1 , the peak is shifted to a_2 . This is useful, since the corresponding speed value s_2 is significantly smaller than s_1 (resp. the yellow final values). Hence, the result of $|v_{d,i}| \cdot |\Delta v_{d,i}$ in Eq. 9 has a smaller magnitude than the original one at initialization.

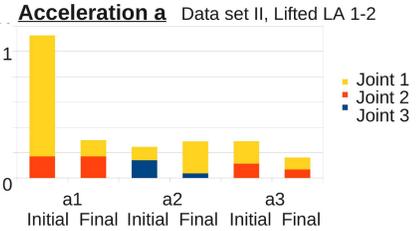


Fig. 9. Acceleration values per joint and per acceleration time step a_x , similarly to Fig. 7. Especially the peak of the joint 1 for the lifted object between LA 1 and 2 is significantly decreased at a_1 . Therefore, the slightly increasing peak of the same joint at a_2 is an acceptable price to pay and can be easily compensated in the overall result of the Objective Function. All acceleration peaks of the other paths of lifted objects in data set II are clearly reduced, too (not depicted).

(Fig. 8). Sometimes, an acceleration peak might be increased slightly to enable a significant reduction of another peak as, e.g., in Fig. 9. The paths of the pushed objects in data set II are not only optimized with respect to dynamics, but also the distance between the desired and the real height above the table are improved (see Fig. 10 and 11).

The change of the path configuration itself is illustrated in Fig. 12 for a path of data set I. It is clearly visible, that the position of the points in the middle of the path has been changed within a large radius (see Section III-C).

The runtime of the method and the stability of the results are good. We want to achieve very accurate start configurations with respect to the desired start position in 3D. Hence, we accept a relatively long runtime for the estimation of a start configuration with about 20 seconds. We use the stochastic optimization approach to be able to compute several different start configurations and to be able to apply the implementation independently of the build-up of the robot. In general, any other procedure to estimate/

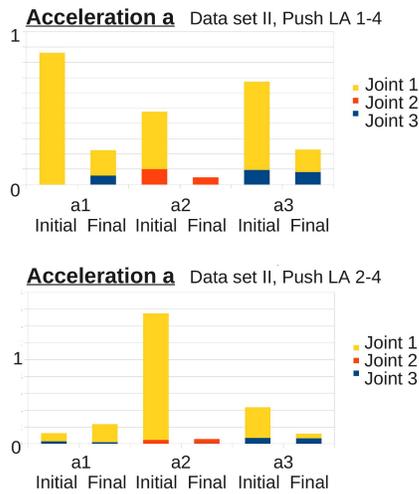


Fig. 10. Acceleration values per joint and per acceleration time step a_x , similarly to Fig. 7. As it can be seen, nearly all of the acceleration peaks along the paths of pushed objects in data set II are significantly reduced. Additionally, the new points are also better positioned with respect to the table plane than the original points, as Fig. 11 shows.

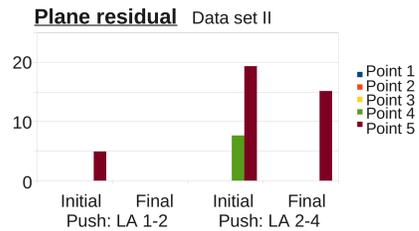


Fig. 11. Remaining residual (in mm) with respect to the table on which the objects are pushed. The residual is shown for each point along the paths of pushed objects in data set II. As it can be seen, the proposed optimization method reduces also the remaining residuals along the path. As described, the residuals are included in the Objective Function. This explains the large magnitude of the values of the Objective Function in Fig. 6 for the paths of the pushed objects in comparison to the paths of lifted objects.

compute the inverse kinematics could be used as well. The optimization of the path itself took less than one second per path. We optimize the paths of the three best start configurations and take the best result after the optimization (Section IV-A). About the half of the other two results is just slightly worse than the corresponding best result (until +1 in the result of the overall Objective Function). The other half has clearly worse results (between +10 and +30 in the result of the overall Objective Function). Since we use the stochastic optimization approach to estimate the inverse kinematics, a repetition of the experiments might lead to slightly varying results. We repeat the experiments three times, hardly any change of the results is observable.

To sum up, we achieve very desirable results in our experiments, since the efficiency in control is significantly improved most of the times.

V. CONCLUSION

We proposed a method for the optimization of path configurations with respect to efficient control. The introduced concept of the Elastic Power Path reflects the elasticity of the path as well as the aim to minimize the necessary energy.

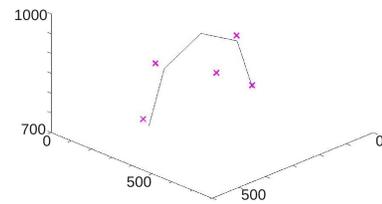


Fig. 12. Exemplary original path (blue) in comparison to the new points (magenta) along the path Lifted LA 1-2, data set I. The possibility to change the position of the middle point at a large extent is clearly used here.

The experiments show, that the number and the magnitude of the peaks in the acceleration profile can be significantly reduced to improve the efficiency. Hence, the system is able to make use of the freedom in path planning which comes along with the abstract representation of tasks.

In future work, further properties stored in the Functionality Map [2] can be included in the optimization. Moreover, the influence of obstacle avoidance could be analyzed.

REFERENCES

- [1] S. Petsch and D. Burschka, "Path configuration for abstractly represented tasks with respect to efficient control," in *IEEE/RSJ IROS: Workshop Beyond Grasping - Modern Approaches for Dynamic Manipulation*, Vilamoura, Algarve, Portugal, 2012.
- [2] S. Petsch and D. Burschka, "Representation of manipulation-relevant object properties and actions for surprise-driven exploration," in *IEEE/RSJ IROS*, San Francisco, USA, 2011, pp. 1221–1227.
- [3] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*. The MIT Press, 2005.
- [4] D. Berenson, T. Simeon, and S. Srinivasa, "Addressing cost-space chasms in manipulation planning," in *IEEE ICRA*, 2011, pp. 4561–4568.
- [5] H. Ding, G. Schnattinger, B. Passenberg, and O. Stursberg, "Improving motion of robotic manipulators by an embedded optimizer," in *IEEE Conf. on Automation Science and Engineering*, 2010, pp. 204–209.
- [6] L. Guangyu, S. Zengqi, and M. Chundi, "Optimal motion planning passing through kinematic singularities for robot arms," in *IEEE/RSJ IROS*, 2006, pp. 4349–4354.
- [7] S. L. Smith, J. Tumová, C. Belta, and D. Rus, "Optimal path planning for surveillance with temporal logic constraints," *IJRR*, vol. 30, no. 14, pp. 1695–1708, 2011.
- [8] I. Belousov, C. Esteves, J.-P. Laumond, and E. Ferre, "Motion planning for the large space manipulators with complicated dynamics," in *IEEE/RSJ IROS*, 2005, pp. 2160–2166.
- [9] T. Rieswijk, G. Brouwn, and G. Honderd, "A robust and efficient approach for the time optimization of path constrained motions of robotic manipulators incorporating actuator torque and jerk constraints," in *IEEE Int. Symp. on Intelligent Control*, 1992, pp. 507–513.
- [10] O. Brock and O. Khatib, "Executing motion plans for robots with many degrees of freedom in dynamic environments," in *IEEE ICRA*, vol. 1, 1998, pp. 1–6.
- [11] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *IJRR*, vol. 21, no. 12, pp. 1031–1052, 2011.
- [12] O. Khatib, "Inertial properties in robotics manipulation: An object-level framework," *IJRR*, vol. 14, no. 1, pp. 19–36, 1995.
- [13] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. of Applied Mechanics*, pp. 215–221, 1955.
- [14] J. J. Craig, *Introduction to Robotics - Mechanics and Control*. Prentice Hall, 2005.
- [15] C. Papazov and D. Burschka, "Stochastic global optimization for robust point set registration," *Computer Vision and Image Understanding*, vol. 115, December 2011.
- [16] O. Ruepp, "Recovery of structure and motion from monocular images under poor lighting and texture conditions," Ph.D. dissertation, Technische Universität München, 2012.