

Incorporating Kinodynamic Constraints in Automated Design of Simple Machines

Can Erdogan, Mike Stilman

Abstract—Robots are inherently limited by constraints on their motor power, battery life, and structural rigidity. Using simple machines and exploiting their mechanical advantage can significantly increase the breadth of a robot's capabilities. In this work, we present an autonomous planner which allows a robot to determine how arbitrary rigid objects in its environment can be utilized in machine designs to overcome physical challenges. First, the designed structure must be sufficient to achieve a task given the input force and torque that can be applied by the robot. Second, the structure must be accessible to the robot given its kinematics and geometry so that it can actually be used to perform the task. The output of our algorithm is the configuration of the design components, the pose of the robot to make contact with the design, and the motor torques needed to actuate it. We demonstrate results with the robot Golem Krang, using levers as simple machines, to overturn 100 kg load and to push 240 kg wheeled obstacle.

I. INTRODUCTION

A robot can exert a limited force to its environment due to constraints on its motor torques, battery power and structural rigidity. However, a variety of tasks require going beyond the physical limitations of oneself by using the available tools at the time. Pushing, raising, and overturning heavy objects are common examples of such tasks in construction and debris removal fields (i.e. search and rescue tasks). People mostly turn to simple machines such as levers, pulleys and wrenches at such times where the work can be accomplished by applying less force over a longer distance and time.

For robots to take part in our daily lives, they need to recognize and adopt the use of simple machines as means to interact with the world. In designing simple machines for a robot, a planner needs to consider both kinematic constraints, such as the arm length in reaching out for objects, and dynamic constraints, such as the maximum force the robot can apply while keeping its stability. The incorporation of such kinodynamic constraints in the purposeful manipulation of the environment is a significant leap towards full autonomy.

The design of a simple machine involves discrete choices for which objects to use and continuous choices for the placement of the objects in the design. Given a set of available objects, our goal is to find a subset for which there is a feasible assignment of configurations that satisfy geometric, dynamic and kinodynamic constraints. The geometric constraints express the contact rules (i.e. lever placed on fulcrum) while the dynamic constraints address force interactions (i.e. lever exerting force to the load). Lastly, the kinodynamic constraints ensure that the system can be driven into action with the physical limitations of the robot [1].

In this work, we present an autonomous planner that chooses which objects to use in the design and configures

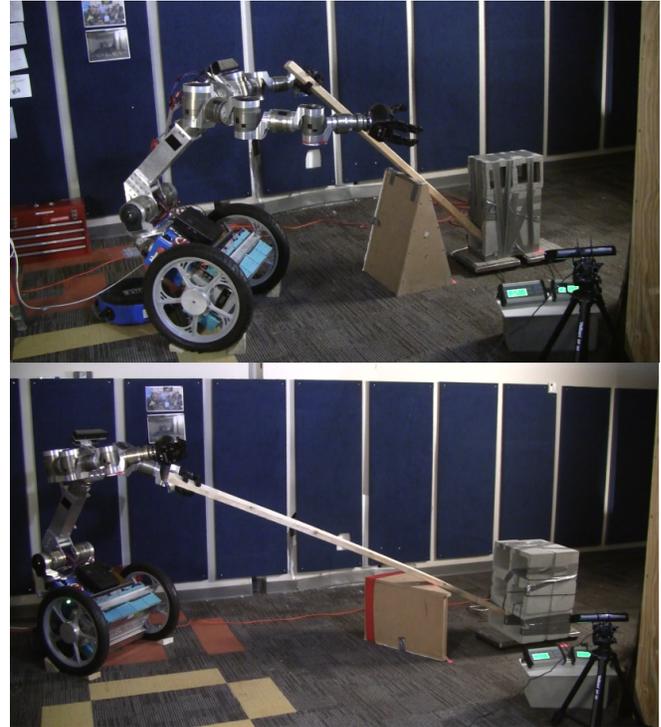


Fig. 1. Golem Krang using 1.7 m and 2.5 m sticks to overturn 50 kg and 100 kg objects respectively with fulcrum repositioned for reachability

them such that they satisfy the constraints of the domain through optimization. The key idea we build upon is to represent the three types of design constraints as generic nonlinear equality and inequality functions within a constraint optimization framework. Then, the set of configurations that satisfy all the constraints, that is a global minimum with zero error, is considered a feasible design. However, if the global minimum of the system is nonzero for the chosen objects, e.g. the lever is not long enough, other object choices are evaluated until all the options are exhausted.

Within this framework, an autonomous planner can search for a design with a large continuous configuration space that is constrained by highly nonlinear functions that represent the domain constraints. Moreover, the representation of domain knowledge as a set of generic constraint functions allows simpler expressions of complex problems such as the manipulation of a heavy load through applying a force to a lever that sits on a fulcrum. Figure 1 demonstrates the output of the proposed planner for two overturning scenarios where the weight of the load is 50 kg and 100 kg respectively.

The robot Golem Krang, designed in Humanoid Robotics

Lab at Georgia Tech, weighs approximately 150 kg, reaches 1.9 m height and actively balances on two wheels for locomotion and manipulation. In Figure 1, at the top, the planner places the fulcrum on its base and dictates Krang where to stand and where to hold the 1.7 m two-by-four object to overturn the 50 kg load. At the bottom, we demonstrate the solution for the second scenario where the planner autonomously chooses a longer lever (2.5 m) for the heavier load at 100 kg and places the fulcrum on its side to ensure that the contact point is still reachable.

In the rest of this paper, we formulate our approach and present results on real world examples. Our contributions can be summarized as follows:

- 1) Expression of simple machine principles as geometric and dynamic constraints
- 2) Incorporation of robot kinematics and dynamics into design parameters in the same nonlinear constraint optimization framework
- 3) The application of the approach to lever designs in overturning and pushing heavy objects

II. RELATED WORK

Manipulation of the environment towards goals has been studied in the artificial intelligence and planning communities extensively. The monkey and banana problem, where a monkey has to move a desk, climb onto it and finally use a stick to reach out to the banana, is a classic example [2]. The earlier work in planning focuses on exhaustive search in the space of discrete and high-level actions, such as “jump to desk” or “pick up stick”. However, for the robotics applications where specific configuration of objects are needed, the search algorithms have been adapted by discretizing the continuous space of configurations either as a grid or based on a domain-specific knowledge [3]. However, such discretization methods can fail to capture solutions at finer resolutions and suffer from large search spaces.

In order to accomplish manipulation and motion planning in continuous spaces, the rapidly-exploring random trees (RRTs) method has been widely adopted [4] where a tree of random samples in the configuration space is grown based on nearest-neighbor principles. A significant extension to this work towards adopting everyday objects has been the accommodation of task-space constraints that limit the motion of objects such as shelves in drawers or doors on hinges [5]. The idea is to improve on the random samples by moving them towards the subspaces that fulfill the task-space constraints using local nonlinear optimization. Local optimization has also been used successfully for grasping arbitrary objects [6], [7]. However, these methods do not address the use of multiple objects to achieve goals.

Recent progress has been made in assembly problems in architecture by using constraint minimization approaches to make design choices [8] and Monte-Carlo sampling methods to design placement of objects in interior decoration [9]. A similar line of thought has also been followed in operations research field where the temporal constraints in a scheduling problem are expressed within an optimization framework

[10]. The key idea is the branch-and-bound approach where the continuous space is partitioned based on discrete choices and then constraint optimization is applied to search for solutions in each subspace [11].

Based on the branch-and-bound approach, we recently proposed a complete planner that constructs functional structures such as bridges and stairs by using blocks in a 2D world [12]. The key idea was to plan in the space of abstract choices of how objects interact with each other (e.g. object A on object B), which effectively partitions the space of configurations, and then solve for the specific configurations of the objects through convex optimization assuming all the constraints are linear. In this work, we extend our framework to capture the complexity of simple machines and limitations of robotic manipulation and incorporate nonlinear constraint optimization to find feasible design solutions.

III. PROBLEM DEFINITION

Let \mathbb{O} be the set of available objects in the environment. The goal of the planner is to choose a subset of the objects to function as a fulcrum and a lever, $\{o_f, o_l\} \subset \mathbb{O}$, and set their configurations $\{x_f, x_l\}$ such that there is a configuration q for the robot Golem Krang where it can manipulate the lever o_l with the joint torques τ to actuate the load object o_l . Before expanding on the design constraints, let us first clarify our assumptions about the capabilities of the robot.

A. Manipulation assumptions

First, we assume that the robot can maintain its initial position $\{x, y\} \in \mathbb{R}^2$ in world coordinates throughout the manipulation of the load. The motivation of this simplifying assumption is the observation that the upper body motion of the robot is often sufficient to actuate the system in numerous experiments and the omission of a timeline, as would be required in full-fledged motion planning, shrinks the number of variables in the problem significantly.

Secondly, we constrain the motion of the robot only to its wheels and the waist joint, and choose not to use the arm joints to apply force. This assumption allows us to guarantee the safety of the joints which maintain their positions with mechanical breaks throughout the motion, and also avoid possible collisions with the body throughout their already limited workspace. Third, we assume that the robot uses its left hand to manipulate the object as opposed to both hands. Figure 2 demonstrates the kinematics of the robot along with the wheel joint θ_1 and the waist joint θ_2 that are actuated.

Lastly, having simplified the robot model as a planar two-link revolute manipulator, we enforce that for the overturning task the robot pushes primarily in the $-z$ direction and for the door opening task in the $+x$ direction. Such an assumption guarantees that a set of torque assignments can be determined for any pose of the robot, except singularities, as opposed to constraining the pose space of the robot significantly to solve for arbitrary directions with a 2D manipulator.

B. Solution space

Let q_b be the degrees of freedom of the base of the robot which are the $\{x, y\}$ position on the ground plane, its

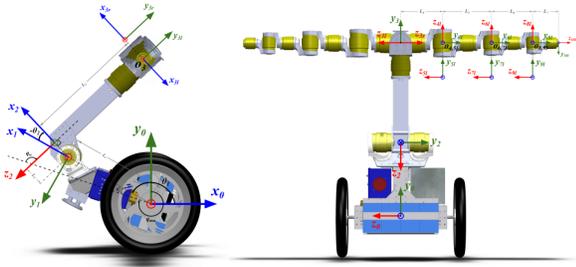


Fig. 2. The kinematics of the robot Golem Krang: The wheel and the waist joints are actuated while the left hand maintains contact with lever.

heading θ , the angle of the base from the vertical θ_1 and the waist joint angle θ_2 . Along with the joints for the 7-dof arm q_l , the robot kinematics comprise 12 variables in total. In addition, the torques $\tau = \{\tau_{\theta_1}, \tau_{\theta_2}\} \in \mathbb{R}^2$ around the wheel and waist joints, and the 6 degrees of freedom for the lever and fulcrum objects, in total, the solution space is 20 dimensional. Note that we incorporate additional variables to simplify the constraint expressions - see Section IV.

C. Constraints

The design of a functional structure imposes geometric and dynamic constraints on its components. The geometric constraints express the contact rules between the different components (e.g. the lever in contact with both the load and the fulcrum), while the dynamic constraints focus on mechanical advantage (e.g. lever arm longer than load arm). Once a design is to be actuated, the user constraints such as reachability and manipulability also need to be considered.

Figure 3 demonstrates a lever design. First, note that we make the assumption that a lever needs to make a line contact (rather than a point) with the edges of the load and the fulcrum in order to ensure the stability of the force interactions. Second, the figure presents the three main geometric constraints: (1) lever in contact with fulcrum, (2) lever in contact with load, (3) fulcrum in contact with the ground along a face. The first two constraints are simple embodiments of what it means to be a lever. The third constraint on the fulcrum pose on the other hand is actually an assumption. It is indeed possible to place a fulcrum on its edge on the ground but it would most probably topple over due to horizontal forces as the lever is moved.

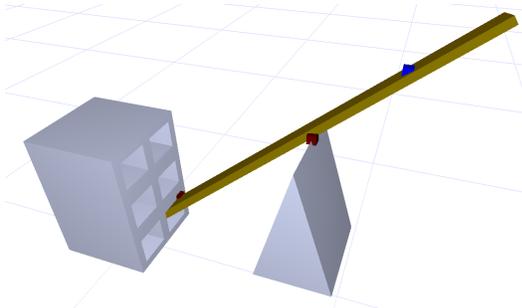


Fig. 3. The design of a lever system with only geometric constraints along with points along the contact edges (red) and the input contact (blue).

In addition to the geometric constraints, the lever design induces dynamic constraints on the configurations of the lever and the fulcrum (we assume the load is fixed). The idea is that the distance from the robot contact point to the fulcrum needs to be more than that of the load contact point to obtain mechanical advantage. Moreover, the force on the load needs to cause sufficient torque, whether it acts against the mass of the object while overturning it or the friction force due to the obstacle in the door opening task. Figure 4 relates the input force f_i , the contact force from the lever to the load, f_c , and the load's gravitational force, f_g , in 2D. Note that we ignore the lever mass (about 1 kg) and the forces parallel to the axis of rotation of the lever.

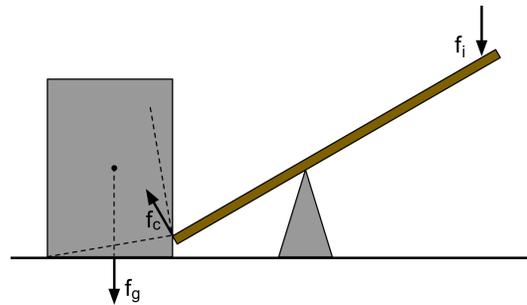


Fig. 4. The forces in interaction in the manipulation of a lever in 2D.

A design that can not be actuated by its user is not functional; and thus, we need to also incorporate the kinematic and dynamic constraints of the robot. First, the robot needs to be positioned such that it can reach the desired contact point. This involves solving for the forward kinematics of the robot, taking into account its link lengths and joint angles to find the hand position. Second, the robot needs to be able to exert the required force using its wheel and waist joints. Here, the simplified two-link planar robot model allow us to use classic force-torque relationship, $\tau = J^T(q)f$, to find the joint torques τ from the robot pose q and input force f . Lastly, the robot needs to be in a balanced pose before it makes contact with the lever. For a wheeled balancing robot like Krang, the center of mass should be over the wheel axis.

Lastly, any design needs to be collision-free in order to be realized. In motion planning, collision avoidance is achieved through sampling the configuration space and rejecting the nodes that lead to collisions. In the approach section, we discuss the heuristics we adopt to simplify the problem.

IV. APPROACH

The goal of the autonomous planner is to choose a subset of the available objects in the environment to construct a lever setup that can be actuated by the robot with kinodynamic constraints. The proposed planner has three major steps: (1) object choices, (2) face-edge matches, and (3) feasibility test. Now, we describe each step in detail.

A. Object choices

To achieve completeness in the space of object choices, the planner evaluates every pair of objects as candidates for the

fulcrum and the lever roles. The idea is to provide the second step these choices as inputs and backtrack in the case of failure, attempting the next pair. For n objects, the exhaustive search requires $n(n-1)/2$ steps; however, some assignments can be eliminated by placing constraints on the size of the objects (i.e. a small cubic object can not be a lever).

B. Face-edge matches

Once the fulcrum and the lever roles are assigned, the design constraints can be applied within an optimization framework to determine the desired object configurations. However, the expression of geometric constraints, such as the lever in contact with the fulcrum, is greatly simplified if the face of the lever and the edge of the fulcrum are predetermined. In this respect, we claim that the partitioning of the configuration space through discrete choices simplifies the infusion of domain knowledge into the system. Moreover, as the optimized functions become simpler, the manifold has less discontinuous and thus, more efficient to trace.

A secondary advantage of incorporating discrete choices in determining the configurations of the objects is the possibility of using domain knowledge to prune the solution space. For instance, once one can make explicit choices about which face of a rectangular prism to choose to make contact, they can also prune the symmetric faces from consideration, thus leading to the elimination of redundant configurations even before consideration. In fact, the detection and exploitation of object symmetries is an active research area [13].

In this intermediary step, the planner essentially builds an abstract design, dictating how the components involved, the load, the lever, the fulcrum, and the robot, come together. In total, five different choices need to be made:

- 1) lever face in contact with some fulcrum edge,
- 2) lever face in contact with the robot gripper,
- 3) lever face in contact with the load edge,
- 4) fulcrum face in contact with the ground (e.g. the base),
- 5) and fulcrum edge in contact with the lever.

Note that we assume a fixed load edge for the lever contact.

Figure 5 demonstrates four example outputs of our planner (without the robot) with different choices for the fulcrum base face, fulcrum edge in contact with the face, and lever face in contact with the load. Given that a rectangular prism lever has 6 faces and a triangle prism fulcrum has 5 faces and 9 edges, there can be 9720 combinations of choices.

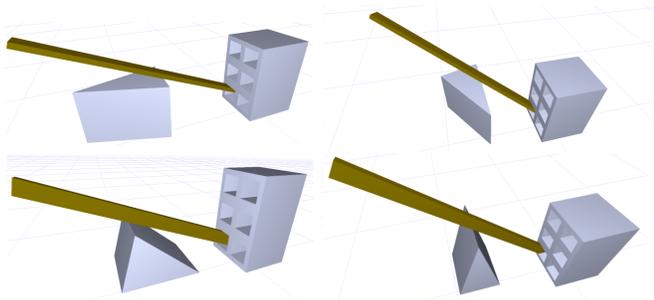


Fig. 5. Effect of different face-edge matches on lever designs.

In closer inspection, we observe most of the options are either replicates (symmetries) or are infeasible because (1) fulcrum edge is on the ground, (2) input or load faces are not perpendicular to the fulcrum face (e.g. can not induce torque), or (3) fulcrum edge is not parallel to the load edge (necessary for the lever to make edge contacts with both objects). These observations shrink the number of feasible choices down to fifteen. However, for more complex objects, heuristics are needed to prioritize favorable choices.

C. Feasibility Test with Constraint Optimization

Once the lever and the fulcrum objects are chosen, and the face-edge assignments that describe the abstract relationship between the objects are made, the feasibility of the overall design can be tested. In this section, we describe our approach to incorporating geometric and dynamic constraints due to the lever design, the robot kinodynamic constraints and collision constraints in evaluating a candidate object choice and coming up with configurations if possible. Next, we expand on two ideas that are used to make the expression of the constraints simpler and the optimization faster.

1) *Latent variables*: In addition to the configuration of the objects and the robot, and the joint torques, we introduce latent variables for: (a/b) arbitrary points on the contact edges of the fulcrum and the load, c_{lf} and c_{llo} respectively, (c) the robot contact point on the lever, c_i , and (d) the magnitude of the force applied along the primary direction, f_m .

The idea behind introducing latent variables is analogous to factoring a complex expression $h(x) = z$ to $f(g(x)) = z$ and then introducing a secondary variable y that is not actually of interest but simplifies the initial constraint by inducing two simpler expressions: $g(x) = y$ and $f(y) = z$. Through experiments, we have seen the increase in the search space of the optimization is worth the ease in expressing knowledge.

2) *Reachability vs. explicit arm joints*: There are two methods to guarantee that the robot reaches the desired contact point. First, one could explicitly solve for the arm joint values, given base joint values, that achieve the desired location for the end-effector. Second, using analytical inverse kinematics [14], check if the given contact point and the necessary end-effector orientation is within the reachable space of the manipulator. In this work, we opt for the second option, allowing us to preserve 7 variables in the solution space for the manipulator arm configuration.

3) *Nonlinear Constraint Optimization*: A common approach in nonlinear constraint optimization is to use the extent of violation of the constraints as the local cost function that is minimized [15]. Subsequently, using random restarts for the initial guess of the correct solution, iterative algorithms such as Gauss-Newton and Levenberg-Marquardt follow the gradient of the cost function to reach a value that satisfies all the constraints. In this work, we use the GTSAM library [16] that allows the expression of nonlinear constraint functions and implements Levenberg-Marquardt optimization.

4) *Incremental Optimization*: Figure 6 demonstrates the factor graph comprised of the unknown variables and the

constraints imposed on them. Each black box is a constraint function imposed on the variables that it is linked to. For instance, at the top left, the fulcrum (F), the lever (L) and a point on their contact edge (C_{LF}) are connected to impose the first constraint that the lever should lie on the fulcrum.

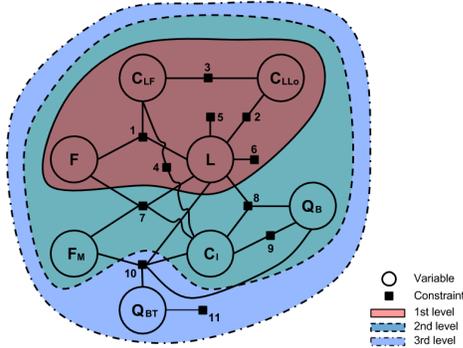


Fig. 6. Incremental optimization for constraints in three levels

Table I summarizes the purpose of the constraints that express the design principles. There are a few interesting ideas embedded in these factors that are worth pointing out. First, the third and fourth constraints that order the contact points define the type of lever design - that is, whether this is a fulcrum-input-load or input-fulcrum-load system. Second, the constraints 5, 6 and 9 express heuristic approaches to avoid collisions where a minimum distance between objects of interest is enforced. Third, the shortcoming of the analytical inverse kinematic approach can be observed in the reachability and balancing factors (8) where the contact point information and orientation from the lever surface needs to be incorporated even though the explicit representation of arm joints would make these constraints simpler.

Number	Purpose
1	Places the lever L on the fulcrum F
2	Ensures that the lever L is in contact with the load
3	Fulcrum contact C_{LF} before load contact C_{LLo}
4	Input point C_I before the fulcrum contact C_{LF}
5	Avoids lever L collisions with the ground
6	Avoids lever L collisions with the load side walls
7	Places contact C_I s.t. load moves with force F_M
8	Reachability and balancing with base joints Q_B
9	Places contact C_I in front of robot
10	Computes the joint torques to induce force F_M
11	Bounds the joint torques Q_{BT}

TABLE I
THE DESCRIPTIONS OF THE CONSTRAINTS

We now discuss the proposed optimization strategy which partitions the process into three steps. The motivation is that when all the variables are optimized together, a large number of guesses may be needed to start a successful optimization. Instead, one can solve for a subset of the variables and use the solution as an initial guess towards solving the entire set.

The proposed solution also intuitively makes sense where the lever and the fulcrum configurations are optimized as the examples in Figure 5, and then the robot contact point and

kinematics are solved, and lastly, once the kinematics are initialized, the joint torques and their limitations are taken into account. Figure 6 displays the three level optimization with the red (solid), green (dashed) and blue (dash-dot) regions. Note that this incremental setup still leads to the nonlinear optimization of all the variables where, for instance, the limits on joint torques affect the fulcrum and the lever poses.

5) *Formulation of the constraints:* With the introduction of latent variables such as points along contact edges, a typical constraint “lever on fulcrum” becomes a set of projection constraints such as: (1) contact point should lie on fulcrum edge, (2) contact point should lie on lever fulcrum face plane, and (3) fulcrum edge points should lie on lever fulcrum face plane. Similarly, in computing the torque due to the robot, the projection of the input point over the force direction and its distance to the axis of rotation is taken. Lastly, for kinodynamic constraints, forward and inverse kinematics, and gripper jacobian J is computed to obtain a measure of error for reachability, balancing and joint torques.

V. RESULTS

The experiments are based on human-robot collaboration where the human provides the perception of available objects in the environment and the realization of the design while the robot conceives the design and is responsible of actuating it. This setup is motivated particularly by search and rescue scenarios where it is preferable that the robot interacts with the heavy loads (i.e. 240 kg) while the human helps with the placement of lighter objects such as the fulcrum and the lever, following the design instructions. Figure 7 displays the planner output as shown in our graphical user-interface and the real-world replication used in the experiments.

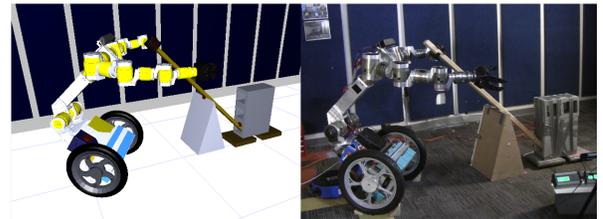


Fig. 7. The ideal planner design and its realization by human collaborator

The main experimental result of our work is the two designs shown in Figure 1 where the planner autonomously chooses the correct levers and places the fulcrum on the appropriate face such that the loads can be overturned with the wheel and waist torques within 150 Nm torque limits. For the 50 kg experiment, the desired input force is 220 N, with 1:2.23 leverage and the desired torques are -67 Nm and 148 Nm. For the 100 kg experiment, the input force is 151 N, with 1:6.50 leverage and the desired torques are -76 Nm and 111 Nm. Note that from these values, we can interpolate that the shorter lever indeed can not be used for the 100 kg experiment without exceeding the torque limits.

To evaluate the accuracy of the actuation of the design, we have accumulated data from three sources: (1) force/torque sensors at the robot gripper, (2) wheel torque sensors, (3)

scales placed under the loads. In Figure 7, the two scales can be seen under the load with the camera recording the values. Figure 8 plots the data from the 50 kg overturning experiment where the object finally topples over just before the 90 second mark. Note that we have marked three points of interest shown in Figure 9b with gray vertical lines: (1) first application of more than 30 N, (2) the load standing on its back edge and (3) the load falling down.

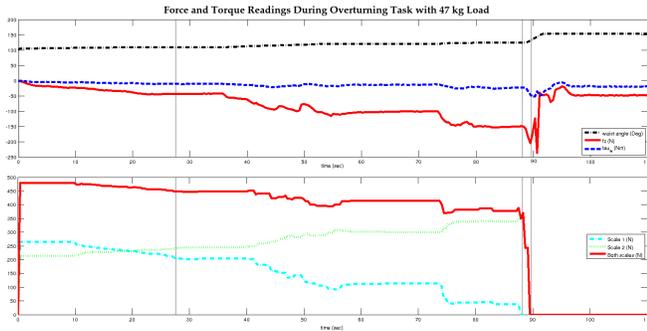


Fig. 8. Force and torque readings throughout the 50 kg overturning task

First, note that the oscillations in z-axis forces and the wheel torques after the fall are due to the robot trying to regain balance while holding the lever. Second, the maximum input force is 205.2 N and the wheel torque is -52 Nm. We propose two reasons why these values are less than expected. First, the planner ignores the forces perpendicular to the axis of rotation of the lever, which can actually help overturn the load. Second, we observe that the waist angle (black, dash-dotted) changes as the input force increases although we assumed the joint angles are fixed. We suspect the reason is the mechanical compliance of the robot, in addition to the flexing of the lever, that causes the upper body to move as opposed to the load. The last observation is the motion of the load center of mass is observed in the decrease in the front scale readings and the corresponding increase in back.

We also conducted an obstacle pushing experiment where a door is blocked by 15 cinder blocks (about 240 kg) on a wheeled platform. Due to the friction with the ground, about 1500 N is needed to move the platform. Figure 9a shows Krang using the 2.5 m lever to open the door using a planner design. Note that we incorporate a new constraint for this experiment to position the fulcrum in contact with the wall to withstand the normal force from pushing the door. Also, the robot was allowed to roll forward as the door opened wider to keep the intended contact point.

Lastly, we analyze the efficiency of our optimization approach as a feasibility test. As emphasized previously, the main challenges are: (1) the commitment to the object choices, (2) the number of face-edge matches and (3) the random restarts. We have shown the number of face-edge matches can be minimized with mesh analysis and we can assume the number of unique available objects is small. However, the random restart approach to avoid local minima persists as the main cause of inefficiency. Despite these challenges, with the three level optimization approach, the



Fig. 9. Key motion steps in pushing and overturning heavy objects

average timings for the overturning and pushing tasks over 20 random load trials are 81.32 and 55.08 seconds. Note that as the required mechanical advantage increases, the feasible subspace shrinks and finding good initializations gets harder.

VI. CONCLUSION

By embedding simple machine principles and the kinodynamic robot constraints in a generic constraint optimization framework, we propose an autonomous planner for object manipulation towards overcoming physical challenges. The planner reasons about object choices, partitions the configuration space of the chosen objects by face-edge matches and uses constraint optimization as a feasibility test. We showed results in lever domains, overturning 50 kg and 100 kg objects and pushing a 240 kg wheeled platform.

In future work, we will expand to autonomous perception and motion planning, tackling different simple machines such as inclined planes and pulleys, and focus on global nonlinear optimization challenges such as efficient initialization routines and early-on detection of conflicting constraints.

REFERENCES

- [1] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *JACM*, 1993.
- [2] C. Green. Application of theorem proving to problem solving. Technical report, DTIC Document, 1969.
- [3] Agarwal M. Mukerjee, A. and P. Bhatia. A qualitative discretization for two-body contacts. In *IJCAI (1)*, pages 915–923. Citeseer, 1995.
- [4] J.J. Kuffner Jr and S.M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *ICRA*. IEEE, 2000.
- [5] M. Stilman. Task constrained motion planning in robot joint space. In *IROS*. IEEE, 2007.
- [6] Hashimoto H. Buss, M. and J.B. Moore. Dextrous hand grasping force optimization. *Robotics and Automation, IEEE Transactions on*, 1996.
- [7] G. Liu and Z. Li. Real-time grasping-force optimization for multifingered manipulation: theory and experiments. *Mechatronics*, 2004.
- [8] P. Geyer. Component-oriented decomposition for multidisciplinary design optimization in building design. *AEI*, 2009.
- [9] Yeung S. Tang C. Terzopoulos D. Chan T. Yu, L. and S. Osher. Make it home: automatic optim. of furniture arrangement. *Siggraph*, 2011.
- [10] Pesch E. Dorndorf, U. and T. Phan-Huy. A time-oriented branch-and-bound algorithm for resource-constrained project scheduling with generalised precedence constraints. *Management Science*, 2000.
- [11] V. Vidal and H. Geffner. Branching and pruning: An optimal temporal poel planner based on constraint programming. *AI*, 2006.
- [12] C. Erdogan and M. Stilman. Planning in constraint space: Automated design of functional structures. *ICRA*, 2013.
- [13] M. Fox and D. Long. The detection and exploitation of symmetry in planning problems. In *IJCAI*, volume 99, pages 956–961, 1999.
- [14] D. Tolani, A. Goswami, and N. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 2000.
- [15] S. Leyffer and A. Mahajan. Nonlinear constrained optimization: methods and software. *Argonne National Laboratory*, 2010.
- [16] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. 2012.