EXPERIMENTS IN CONSTRAINED PREHENSILE MANIPULATION: DISTRIBUTED MANIPULATION WITH ROPES

Bruce Randall Donald
Larry Gariepy
Daniela Rus

Abstract This paper describes our experiments with a distributed manipulation system. We study a system in which multiple robots cooperate to move large objects such as furniture and boxes using a constrained prehensile manipulation mode, by wrapping ropes around them. The system consists of three manipulation skills: tieing ropes around objects, affecting translations using a flossing manipulation gait, and affecting rotations using a ratcheting manipulation gait. We present experimental data and discuss the non-holonomic nature of this system.

1 INTRODUCTION

We are working on distributed manipulation problems, in which two or more robots apply spatially-distributed forces to rearrange the objects in their work space; the computation is distributed among the robots so as to qualify as a distributed computation. In our previous work Donald et al. 1994; Rus et al, 1995; Donald et al., 1997a; Donald et al., 1997b we presented several distributed manipulation protocols for non-prehensile pushing and reorienting of large objects, where the robots have no models of these objects, there is no explicit synchronization between their actions, and there is no explicit communication between the robots. In this paper we describe a new class of distributed manipulation algorithms in which multiple mobile robots manipulate large objects using ropes.

We are implementing a system in which multiple robots cooperate to move large objects such as furniture and boxes using a constrained prehensile manipulation mode, by wrapping ropes around them. In our system, two RWI B14 robots are linked by a long rope (5 meters) which is tied around the "waistline" of the robots. This system can rotate and translate objects by wrapping the rope around them and pulling on the rope in a coordinated fashion. The rope acts as both coupling and indirect communication medium in this system, much like the hands of a dancing couple.

Manipulating objects with ropes has several advantages over the direct manipulation of an object using a team of robots Rus et al, 1995; Donald et al., 1997a; Donald et al., 1997b. First, the method allows for the parallel and synchronized manipulation of multiple objects. (That is, several objects can be wrapped together). Second, wrapping a rope around an object permits *constrained prehensile manipulation* with non-prehensile robots. The rope can conform to any object geometry on-line, in that the same wrapping protocol works for all geometries and there is no need to specify a model in advance. Finally, the rope can be viewed as a tool that allows the robot system to exert torques that are larger than the torque limit of each individual robot. Our notion of constrained prehensile manipulation is different than the classical definition of prehensile manipulation, which usually denotes a force closure grasp. In *constrained prehensile manipulation,* perturbations along any differential direction can be resisted (in our case, due to the taut rope) but the object can be moved only along certain lower dimensional degrees of freedom. Thus, constrained prehensile manipulation systems are *non-holonomic.*

1.1 INSPIRATION: BIOMIMETIC DESIGNS

This approach to distributed manipulation is inspired by a class of biological systems, including flagella and cilia. Many bacteria possess flagella – that is, long, protein-containing hair-like appendages that propel the bacteria through liquid media or across solid surfaces, usually toward a desired nutrient or away from a deleterious agent. In biological systems, these thin helical filaments self-assemble from 53-kd sub-units of a single protein, flagellin. An *E. coli* or *S. typhimurium* bacterium has about six flagella, which are about 150 Å in diameter and 10 μm long. Bacterial flagella are much smaller and far less complex than eucaryotic flagella and cilia. Artificial flagella and cilia could, in principle, permit robot systems to combine structure and function, and allow dynamic and automatic reconfiguration in response to environmental challenges. Such physical mechanisms allow unit modules to dynamically and automatically configure themselves into more complex robots, or for more complex tasks (for example, manipulation vs. locomotion). In particular, aggregations of flagella or cilia may be able to self-organize into the optimum configuration for a manipulation or locomotion task. Artificial *cilia* with active control have already been demonstrated for distributed manipulation tasks on the micro scale, by our group and others Böhringer, Donald, and MacDonald, 1996; Böhringer et al., 1997a; Böhringer et al., 1997b. Artificial *flagella* are more difficult to construct. As a first step towards exploring the power of distributed manipulation using artificial flagella, we have built a system of distributed robots employing synthetic *passive* flagella, which we term "ropes." These experiments allow us to "factor" the power gained in a flagellary system into two components: (1) the passive kinematics of the ropes/flagella, and (2) the additional actuated degrees of freedom gained by active control of the flagella. In this paper, we explore the power of (1), using entirely passive flagella/ropes. Note, however, that the flagella/ropes can move and effect manipulation when their "parent" robots move; however, they cannot be independently actuated. Thus, our work explores a new kind of underactuated distributed manipulation system based on a simple biomimetic model. As we will see later, this system may be encoded and analyzed as a non-holononmic system for distributed manipulation.

2 DISTRIBUTED MANIPULATION WITH ROPES

Consider the task whose goal is to change the pose of a large box, or a group of boxes. A team of robots can use non-prehensile strategies such as those described in Rus et al, 1995; Donald et al., 1997a; Donald et al., 1997b to accomplish this task. A different approach is to use a constrained prehensile strategy. Because the objects we wish to move are large, it is impractical to envision robots equipped with big enough grippers to do such manipulation tasks. Instead, we focus on strategies that augment the basic architecture of mobile robots with simple tools. In this section we describe an approach where the robots can achieve a grasp on large boxes or collections of objects by wrapping the objects with a rope (independent of the overall geometry). We then explore cooperative strategies for manipulating an object wrapped with a rope. Our experimental infrastructure is described in Section 2.1. Section 2.2 describes a system that implements such strategies. Section 3 describes our experimental results with this system. Section 4 describes some interesting issues related to the geometric foundations of motion control Ostrowski and Burdick, 1995 in the context of a mechanical analysis for these motions.

2.1 EXPERIMENTAL INFRASTRUCTURE

Our distributed manipulation team consists of three mobile robots: two RWI B14 robots equipped with sonar, discrete contact sensors, IR, and radio modems; and one RWI Pioneer robot equipped with sonar, a radio modem, and a parallel-jaw gripper. The B14 robots are named Bonnie and Clyde and the RWI Pioneer robot is named Ben. We augmented the basic hardware of these robots by using a long rope (over 5 meters) which is tied at one end to Bonnie and at the other end to Clyde. We incorporated a tool in Ben's gripper that extends the effective width of the gripper from 2 cm to 15 cm.

2.2 A DISTRIBUTED SYSTEM FOR MANIPULATION WITH ROPES

The distributed manipulation system consists of three robots: Bonnie and Clyde, which are linked by a long rope (5 meters) tied around their "waist-lines," and Ben, who is free to move and carries a rope grasping tool. When an object is wrapped and constrained by the rope, we say it is *bound*. Bonnie and Clyde are unable to bind an object by themselves; therefore Ben acts as an "enzyme" to catalyze the binding "reaction."

The distributed manipulation system developed using the three robots consists of three skills for manipulating objects with ropes: tieing a rope around a large object or a collection of objects, effecting rotations on a bound object, and two ways of effecting translations on a bound object. The control for each skill is implemented as a protocol using active and passive robots. We now detail the components of this system.

Binding: Wrapping the rope around an object. The goal of this skill is for a team of two robots to surround an object with a rope to generate a *rope grasp*. The ends of the rope are tied along the "waistline" of each robot. The basic idea of the algorithm is to keep one robot stationary while controlling the other robot to move around the object with the rope held taut. At some point, the moving robot has to cross over the rope in order to complete the grasp; this step requires coordination with the stationary robot.

We have developed a control algorithm for this skill and implemented it using two RWI B14 robots and one Pioneer robot (see Figure 1). In the first phase, the B14 robots are on the same side of the box. Clyde, the active robot (the right robot in Figure 1) moves so as to keep the rope taut around the box. Bonnie, the left robot in Figure 1, is passive and stays in place. The control for encircling the box is implemented as hybrid control of a compliant rotation. The robot starts with a calibra-

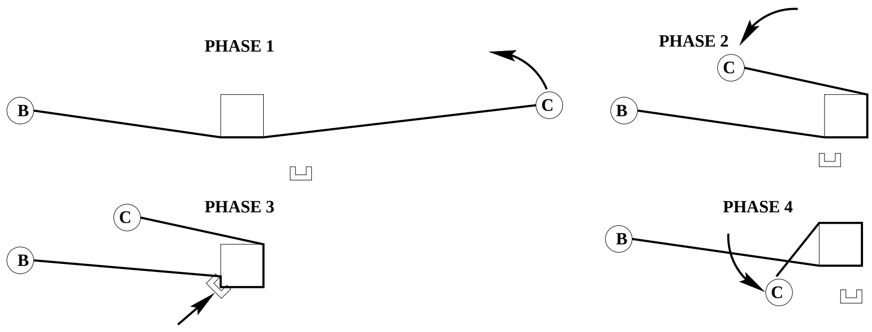

Figure 1 The 4 phases of binding (wrapping a rope around a box). The RWI robots are denoted by disks. Bonnie is denoted by B; Clyde is denoted by C. The Pioneer helper is denoted by the gripper icon. Translation: From the state obtained in Phase 4, translations can be effected by having Bonnie and Clyde move with parallel, synchronous velocities. For example, from Phase 4, the box can be translated 'southwest' by commanding the same motion to B and C. For pure translations, B and C should be equidistant from the box. That can be accomplished using a ratchet (regrasp) step.

tion step to determine the level of power necessary to achieve rotational compliance to the box. The robot tries to move in a polygonal approximation to a circle. Due to the radial force along the rope, the robot's heading complies to remain normal to the radial force, thereby effecting a spiral. The robot continues this movement until it gets close to having to cross the rope. The following function implements a half-tour of the box:

```
(define (half-surround-box)
  (rotate-power (calibrate 1 200))
  (let ((initial-heading (rotate-where)))
    (set-translate-velocity! 200)
    (let repeat ()
         (translate-relative 100)
         ;Stop after 165 degrees
         (if (and (= 0 (contact-mode))
                  (< (abs (- initial-heading
                             (rotate-where))) 165))
             (repeat)
             (stop)))))
```

The first phase ends when the encircling robot has completed a 165 degrees tour around the box. In the second phase, Clyde crosses the rope. The algorithm requires Bonnie to drop the rope to the ground to enable Clyde to drive over the rope. A simple forward movement of Bonnie would cause the rope to drop to the ground, but it will break the grasp. Furthermore, the inverse move is not guaranteed to reestablish the grasp because the rope can slide under the box. Thus, it is important to maintain the partial grasp on the box during the crossover. Our approach is to use a Pioneer helper robot, that can support the rope while the passive robot gives slack.

The algorithm for the second phase starts with Ben locating the corner of the object closest to Bonnie. Ben uses its gripper to push the rope against the box, thus securing the rope grasp. Then Bonnie gives some slack in the rope. In the third phase, Clyde crosses over the rope. In the fourth phase, Bonnie tightens the rope. The tension in the rope signals the helper to let go and move away.

Translating a bound object. In the translation skill, two robots translate a bound object by moving with parallel, synchronous velocities (see Figure 1). The basic idea of this algorithm is to start by regrasping a bound object so that (1) the two robots are equidistant from the box along a line called *the rope line* and (2) the rope line is perpendicular to the desired direction of translation. These two properties are achieved by using the active part of the wrapping algorithm: keeping the rope taut, both robots rotate counter-clockwise about the box until the rope line becomes perpendicular to the desired direction of translation. At this point, a regrasp step is necessary to ensure that both robots are equidistant along the new rope line. From this position, the robots can translate by commanding synchronous velocities.

Flossing: rotating a bound object. In the flossing skill, two robots rotate an object in place by coordinating movements of the rope between them. This type of manipulation is similar in flavor to the pusher/steerer system described in Böhringer et al., 1997c. Flossing is preceded by a binding operation. We have developed a control algorithm for this skill and implemented it using two RWI B14 robots (see Figure 2). In the first phase the active robot translates forward. The passive robot follows along, feeling force F_p and keeping the rope taut. This action causes the object to rotate. This phase terminates when the passive robot stops in place, thus increasing the tension in the rope. The active robot senses this impediment and stops. At this point, the robots can reverse the passive/active roles and repeat the first phase. When Bonnie is the active robot, the object rotates clockwise; when Clyde is the active robot, the object rotates counterclockwise.

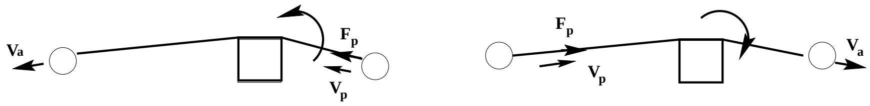

Figure 2 The 2 phases of rotating an object by "flossing." The robots are denoted by disks. V_a denotes the velocity of the active robot. V_p denoted the velocity of the passive robot. F_p is the force felt through the rope by the passive robot.

Ratcheting: translating a bound object. In the ratcheting skill, two robots translate an object by coordinating movements of the rope between them. As in the flossing skill, this type of manipulation is also preceded by a binding operation. The manipulation system may also combine ratcheting manipulation with flossing manipulation.

We have developed a control algorithm for this skill and implemented it using two RWI B14 robots (see Figure 3). The algorithm for translations iterates the following phases. In the first phase the active robot translates forward with velocity v_a. The passive robot follows along feeling F_p and keeping the rope taut. This action causes the robot to translate. When this phase is executed for the first time, it terminates with the passive robot sensing contact with the object. This is called the *calibration step*. During a calibration step, the passive robot measures the approximate distance to the object. In subsequent repetitions of this phase, the passive robot uses dead reckoning to decide when to stop.

In the second phase the robots decrease the tension in the rope and translate together in the opposite direction. The object does not move in this phase and the rope slips against the object. The effect of this phase is to re-grasp the object so as to move the active robot closer to the object.

2.3 DETAILS OF MANIPULATION ALGORITHMS

All code for our system is publically available on the WWW; send email to {`brd,rus`}`@cs.dartmouth.edu` to obtain the source.

Experimental Setup. Each experiment begins with the two B14 robots, Bonnie and Clyde situated on opposite sides of a box, with a rope strung between them. The helper robot, Ben, is situated next to the side of the box closest to the rope.

The Binding Algorithm. The binding phase begins with Clyde performing a calibration routine. Let F_p be the force on the passive robot (Bonnie) and F_a the force on the active robot (Clyde). In order for the initial rope tension not to affect the calibration step, Clyde first turns 90 degrees to face toward Bonnie, and moves forward to release all tension in the rope ($F_a = 0$). He then measures the amount of rotational power necessary to achieve rotational compliance.[1] Ready to begin, he now returns to his original position and orientation.

Clyde applies the rotational power he just computed, and begins to move forward. As he moves forward, the tension in the rope combined with Clyde's state of rotational compliance causes him to turn and follow a (piecewise) circular path around the box. The tension in the rope stays roughly constant ($F_a = 4$ lbs) and Clyde monitors his odometry so that once his torso has rotated through 180 degrees, he will stop.

At the point he stops, he should be less than two feet from the length of rope running from Bonnie to the box. Now, he waits (for a predetermined amount of time, about 30 seconds [2]) and the action shifts to Bonnie and Ben.

Through this first phase of the experiment, Bonnie and Ben have been dormant,[3] but now they come to life (using timing delays: there is no communication that occurs between Clyde and the other two robots at this point). Bonnie turns toward the box, and enters a state of translational compliance. Ben seeks out the corner of the box nearest to both the rope and to Bonnie. Once he locates the corner, he moves out and aligns himself with the box diagonal passing through that corner. He is now ready to grasp the rope against the box, and he slowly begins moving toward the corner.

As Ben's left claw encounters the rope, it exerts a force upon the rope that drags Bonnie a little way forward. However, at this point he must stop to avoid a motor stall (he is alerted to his contact with the rope using his touch sensors). When Bonnie notices that she has moved, she takes this as a signal from Ben that he is almost in position. At first, she gives just a little bit of slack ($F_p = 1lb.$). This allows Ben to get in position against the box once he re-engages his motors. After waiting a couple of seconds, Bonnie assumes that Ben is in position, and she moves forward some more. This time, the segment of rope between Ben and Bonnie falls to the ground. Note that for the crossover phase, F_a and F_p are not equal! F_a is still about 1 lb., whereas F_p is 0 lbs.

Now the stage is set for Clyde to actually cross over the rope. When his timeout expires, he once again goes to a state of rotational compliance, and moves forward for another 45 degrees worth of rotation. He then goes to a dormant state again, waiting for Bonnie and Ben to tidy up. When Bonnie wakes up, she backs up to retension the rope (so that once again, $F_p = F_a = 4$ lbs). This has a side effect of signaling to Ben that his job is done. The tension in the rope actually moves him backward a little bit. When he detects this movement, Ben backs up and moves out of the way so that Bonnie and Clyde can manipulate the box.

In order to perform flossing and ratchet tasks, Bonnie needs to move around to the opposite side of the box from Clyde. So she now goes to a state of rotational compliance, and moves forward for about 135 degrees worth of rotation. When she reaches the final position, she rotates 90 degrees to face the box. Clyde does the same, and the two robots are now ready to perform manipulation tasks with the box.

Ratcheting. The ratchet algorithm translates an object. However, one side effect of the ratchet algorithm is that the box also rotates as it translates.

The setup for ratcheting assumes that a wrapping operation has been performed, and that the robots are positioned on opposite sides of the box, facing each other (Figure 4). For a ratchet operation, we designate one robot to be active, and the other to be passive. The active robot will do the work of moving the box, while the passive robot's main job is to help maintain the "grip" on the box with the rope. Initially, the rope is at moderately high tension ($F_a = F_p = 8$ lbs).

In order for the robots to be in synch, the passive robot begins the experiment by signaling the active robot that it is ready to begin. The signal that is used is a small tug on the rope – just enough for the active robot to notice itself moving. The active robot begins the experiment by sitting in a loop that monitors the odometer, while maintaining a state of translational compliance. When it notices movement, it switches into its active role.

After giving the signal, the passive robot goes into a state of translational compliance. The active robot begins towing both the box and the passive robot (at this point, we have $F_a = F_p =$ [*friction resistance of the box*] + [*resistance of the passive robot*] which in practice is about 15 lbs). The active robot will continue pulling until it senses the stop signal from the passive robot. During the first iteration, this occurs when the passive robot senses with its bump sensors that it is touching the box. To signal the active robot, the passive robot "digs its heels in" by applying significant power to its translational motors in the direction opposite to the direction of motion ($V_p \rightarrow 0$). The effect of this is to slow down the active robot (F_a reaches its maximum value at this point). While the active robot is pulling the box, it is also monitoring its velocity (V_a). When V_a drops below a specified threshold (we use approx. $.65V_{max}$, where V_{max} is the velocity with which the robot was commanded to move) the active robot assumes it is receiving the stop signal, and it stops.

The passive robot has now gone as far forward as it can with the length of rope it was given. It is time to "ratchet." In order for the rope to slide around the box (as opposed to moving the box), the tension in the rope must first be minimized. This is easily accomplished by having the active robot begin to move back toward the box immediately after receiving the stop signal, but having the passive robot wait for a short time (about 1.5 sec) before moving away from the box. During this phase, $F_p = F_a < 1lb$. Now the roles are pretty much reversed. The active robot monitors its bump sensors to watch for contact with the box, and the passive robot moves, towing the rope (which slips along the box), while waiting for the stop signal from its partner. Again, the stop signal is transmitted through the rope. However, one important difference is that the active robot is not in a state of translational compliance. It is moving of its own volition, as is the passive robot.

Once the active robot reaches the box, the first iteration of the ratchet task is complete. Further iterations can now be executed, each one moving the box further along its path. We decided to implement one minor change for the subsequent iterations, however. Rather than having contact with the box determine when the stop signal was issued, we preferred instead to have the robots measure the length of rope they have to work with during the first iteration, and use that measurement during later iterations, so that it was not necessary to bump into the box every time. This gives the experiments a smoother feel.

Flossing. The flossing algorithm reorients an object. The task itself does not actually effect a pure rotation, but also translates the box slightly as a side effect. To demonstrate reorienting the box in either direction, we have the robots perform a kind of tug-o-war over the box. The algorithm for this task is symmetric with respect to the two robots. They simply take turns towing the box (and each other), and rotating the box back-and-forth. The robots take turns as the active and passive members of the experiment. We use a stop signal just like the one in the ratchet experiment to allow the passive robot to signal the active robot that it is time to switch roles.

Figure 3 The 2 phases of translating an object with a ratchet motion. The robots are denoted by disks. V_a denotes the velocity of the active robot. V_p denoted the velocity of the passive robot. F_p is the force felt through the rope by the passive robot.

3 EXPERIMENTS

We have implemented the distributed manipulation skills described in Section 2.2 using the robots described in Section 2.1. The RWI robots are programmed in MOBOT-SCHEME Rees and Donald, 1992 and the helper Pioneer robot is programmed in C++. Figure 4 show some snapshots from a typical run.

Figure 4 Six snapshots from an execution of binding with ropes. The top-left figure shows the initial configuration for the binding experiment. The top-middle figure shows the robot Clyde encircling the box. The top-right picture show that Clyde has come halfway around the box and Ben is ready to help with the knot tieing component of the experiment. The bottom-left picture shows that Ben has traveled to the box corner and is currently holding the rope. In the bottom-middle picture Clyde crosses over the rope. In the bottom-right image the box is wrapped with the rope. We have also implemented the translation motion (Figure 1), the ratchet motion (Figure 3) and the flossing motion (Figure 2) once the object is bound.

Figure 6 shows reliability data for some recent experiments run with this system. The failures of the binding skill were related to sensor

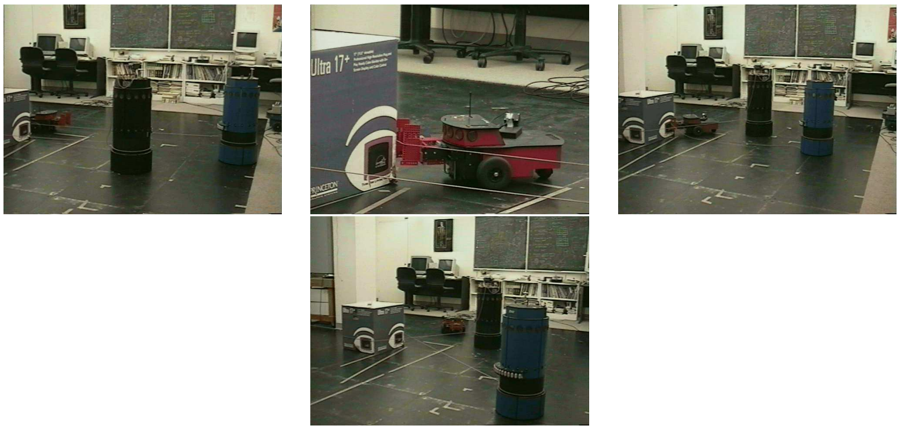

Figure 5 Details of a rope binding experiment. The first picture show that Clyde has come halfway around the box and Ben is ready to help with the knot tieing component of the experiment. The second picture shows that Ben has traveled to the box corner and is currently holding the rope. In the third picture Clyde crosses over the rope. In the fourth image the box is wrapped with the rope.

inaccuracies. The majority of the observed errors were due to the fact that Clyde, the active robot during binding, uses odometry to terminate the first phase of the algorithm. This method introduces an error of up to 3 feet in the positioning of the robot relative to the rope and leads to the failure of the rope crossing step. The rest of the errors were due to the helper robot, Ben, who uses odometry to locate the corner of the box. Occasionally, Ben misses the corner and thus does not get a good grasp of the rope. The failures in flossing and ratcheting were due to signaling errors. Occasionally, the passive robot stops prematurely. We believe that these errors are due to low battery levels during those specific experiments, but more investigation is necessary.

4 ANALYSIS

In this section we provide a brief analysis of the manipulation grammar and the non-holonomic nature of the rope manipulation system.

4.1 MANIPULATION GRAMMAR

It is possible to develop a simple manipulation grammar Böhringer, Donald, and MacDonald, 1996 to describe our manipulation system. The manipulation algorithms described above provide four basic skills:

Task	Successes	Tries	Reliability
Binding	29	40	72.5 %
Flossing	19	21	90.5 %
Ratcheting	19	26	73 %
Transitions	8	11	72.7 %

Figure 6 This table contains reliability data for our experiments with binding, and for manipulation by flossing and ratcheting. The last row shows data for transitions from binding to flossing and from binding to ratcheting.

B. *Binding* (Sec. 2.3).

T. *Translation* (Fig. 1, post-Phase 4, and Sec. 2.3),

O. *Orientation* (Sec. 2.3), and

G. *Regrasping* (when the rope slips along the box in Sec. 2.3).

Let us call these primitives B, T, O, and G, respectively. If O is an orientation step, let O^R be the *reverse* step, performed by reversing the robots' directions. In this case, flossing can be written as $(O, O^R)^*$, the repeated iteration of O followed by O^R. Note that no direct communication is required to switch between O and O^R; the signal is transmitted through the task (the rope).

Similarly, the ratcheting operation can be written as $(O \mid T, G)^*$. Again, no direct communication is required to switch between regrasp and orientation/translation steps. To summarize:

1. *Flossing* $= (O, O^R)^*$.
2. *Ratcheting* $= (O \mid T, G)^*$.

All operations must be preceeded by a binding step B. For pure translations, Bonnie and Clyde should be equidistant from the box. That can be accomplished by inserting a regrasp step G. Thus, a simple translation could be accomplished by (B, G, T).

4.2 NON-HOLONOMIC MANIPULATION SYSTEMS

In mechanics, systems may be *holonomic* or *non-holonomic*. In general, a *holonomic constraint* is a wholly integrable sub-bundle E of the tangent bundle. The outcome for a non-holonomic system is path-dependent. Non-holonomic systems have been studied in robotics. Examples include: car-like robots, tractor-trailers, bicycles, roller-blades, airplanes, submarines, satellites, and spherical fingertips rolling on a manipulandum. In robotics, a non-holonomic system is usually defined by a series of non-integrable constraints of the form $\Gamma_i(p, v) = 0$ on the tangent bundle Baillieul, Brockett, Donald, et al., 1990; Latombe, 1991; Barraquand and Latombe, 1993; Goldberg et al., 1995; Latombe, 1993. For example, whereas holonomic kinematics can be expressed in terms of algebraic equations which constrain the internal, rotational coordinates of a robot to the absolute position/orientation of the body of interest, non-holonomic kinematics are expressible with differential relationships only. This distinction has important implications for the implementation of a control system.

It is well known (see Figure 7) that many underactuated manipulation systems are naturally modeled as non-holonomic systems. Figure 7 illustrates a *one-sided* non-holonomic constraint: R can push O but not pull it. Therefore only 'northeast' diagonal pushing motions are possible. On the other hand, if R could *pull* O, then diagonal pushing motions in either direction (northeast or southwest) are possible. This would define an *isotropic nonholonomy.*

Our system admits an isotropic non-holonomic model as follows: Let R_1 and R_2 be the positions of Bonnie and Clyde (respectively), and let r be one half the diameter of the manipulandum (for a sensible definition of the *diameter* of a polygon, see Goldberg et al., 1993). Although in general R_i is two dimensional, for the sake of illustration, assume that $R_1, R_2 \in R$. With the rope taut, the distance between R_1 and R_2 may be closely approximated as fixed once the box is wrapped. Thus the combined, simultaneous configurations of R_1 and R_2 may be collapsed into a single parameter $d \in R$, representing the *rope position.* Finally, let θ be the orientation of the box, and x its translational position.

Now, let us consider a single orientation step O (Figure 8-left). When the robots move to manipulate the box, it translates and rotates in a non-linear fashion that depends on the box geometry (Figure 8). For the orientation task, the progress this strategy makes is the average slope of the linearization of this curve in d-θ space. This slope is approximately $(2\pi r)^{-1}$.

Non-holonomic analyses can also be done for translation (T), flossing $(O, O^R)^*$, translational ratcheting $(T, G)^*$, and orientational ratcheting $(O, G)^*$. (Figures 8, 9, and 10).

By interleaving regrasp steps with translation steps, the box can be translated repeatedly; the motion in configuration space is like a 'ratchet'

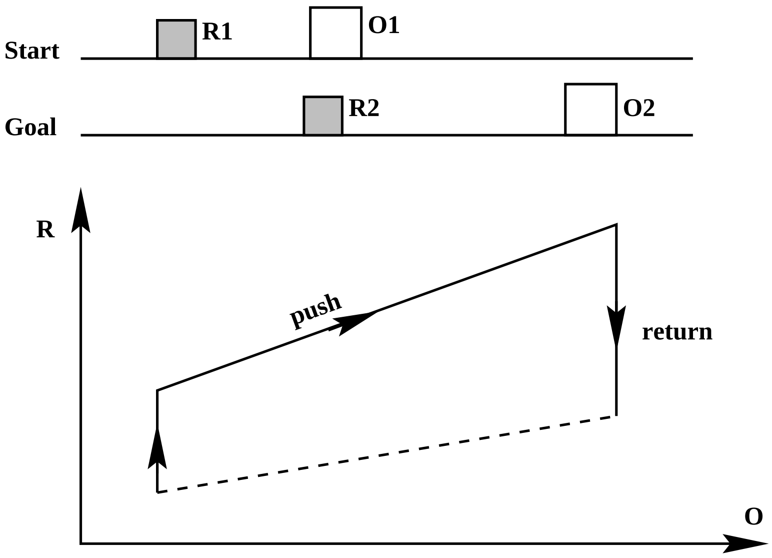

Figure 7 An underactuated manipulation system may be modeled as a non-holonomic system (simplified example from Donald, 1989[pp.40-42,260-271], based on a suggestion of Koditschek). Here a robot R starting at R_1 must push an object O from configuration O_1 to O_2. The goal state of the robot is R_2. Both R and O have one-dimensional configuration spaces, so their combined configuration space is R^2. If the system were fully actuated, then R could move from R_1 to R_2 while O simultaneously moved from O_1 to O_2 (dotted path). Since O is not actuated, R must first move to O_1, then push O from O_1 to O_2, and finally return to R_2 (solid path). Not all degrees of freedom (DOF) are available to R (for example, diagonal pushing paths are only possible when R and O are in contact). However, by a series of maneuvers along the reduced differential DOF, higher DOF strategies may be carried out. This is the essence of a non-holonomic system.

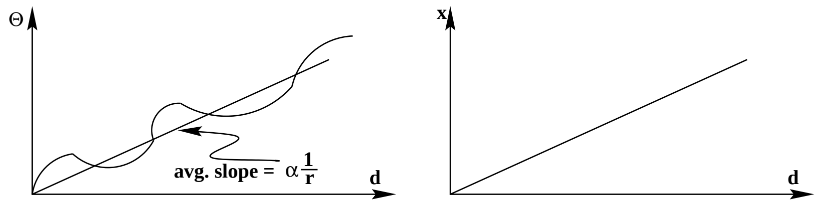

Figure 8 (Left) Non-holonomic Model of a single Orientation Step O. The orientation θ of the box as a function of the control parameter d, modeled in the non-holonomic configuration space $R \times S^1$. This figure shows one half (O) of an orientational flossing cycle (O, O^R). (Right) Non-holonomic model of a translation step T. x represents the position of the box.

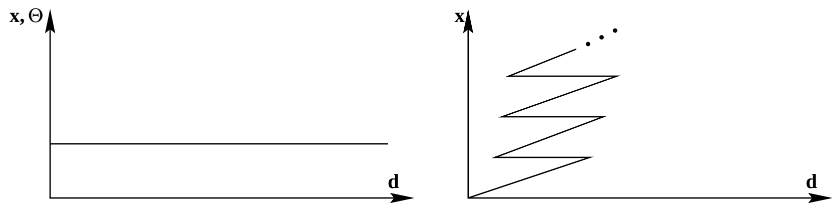

Figure 9 (Left) Non-holonomic Model of a Regrasp Step G. When the rope is slightly slack, it slips along the box without affecting its pose. (Right) The translational ratchet. Non-holonomic model of interleaved translational and regrasp steps, $(T, G)^*$.

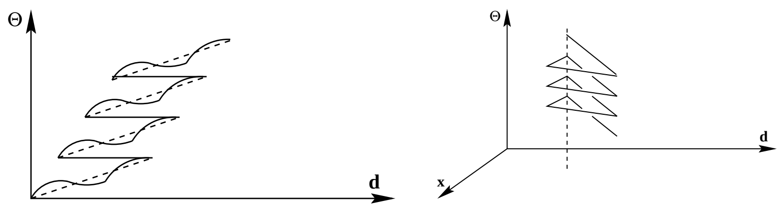

Figure 10 (Left) The orientational ratchet. Non-holonomic manipulation gait of interleaved orientational and regrasp steps $(O, G)^*$.
(Right) The 3D ratchet in the configuration space $R^2 \times S^1$ is a non-holonomic manipulation gait composed of interleaved translational, rotational, and regrasp steps $(O, T, G)^*$.

(Figure 9). Note that not all of the degrees of freedom (DOF) in the configuration space are accessible; only a 1D subset of the DOF are differentially possible. Therefore, this kind of ratchet maneuver must be employed to access all the DOF. Note that the ratchet maneuvers are similar to parking maneuvers common in non-holonomic motion planning for wheeled vehicles Latombe, 1991; Barraquand and Latombe, 1993; Goldberg et al., 1995; Latombe, 1993. A similar picture is obtained by interleaving regrasp steps with orientation steps, except the orientation steps are non-linear (Figure 10). By interleaving all three kinds of steps (translational, orientation, and regrasp), arbitrary (x, θ) poses of the box can be achieved. This corresponds to a non-holonomic ratchet in the 3-dimensional configuration space $R^2 \times S^1$ parameterized by (d, x, θ) (Figure 10). Both 2D and 3D ratchets can be viewed as manipulation gaits in a non-holonomic control space.

5 SUMMARY

In this paper we described the constrained prehensile manipulation of an object bound by a rope. We presented four algorithms for the distributed manipulation of a roped object: binding (that is, tieing a rope around an object), translating, flossing (that is, effecting rotations of a bound object), and ratcheting (that is, effecting translations of a bound object). Our algorithm use a team of three robots that do not communicate directly but use the tension in the rope as a signaling mechanism. The role of the three robots employed by our systems can be separated into: active, passive, and helper. The algorithms consist of phases that can be repeated by switching the roles of the active and passive robots. We have also discussed briefly some interesting issues related to the geometric foundations of motion control in the context of a mechanical analysis for these motions.

We implemented the three manipulation skills and experimented with a distributed system of three robots that can bind an object, rotate a bound object in place, and translate a bound object. In our system, the robots can also make transitions between these skills: a binding operation can be followed immediately by a flossing or ratcheting operation. Flossing and ratcheting can also be alternated. Our experimental results show levels of reliability between 72.5 % and 90 % for the manipulation skills.

Acknowledgements

This paper describes research done in the Dartmouth Robotics Laboratory. Support for this work was provided in part by the following grants from the National Science Foundation: NSF CAREER IRI-9624286, NSF IRI-9714332, NSF IIS-9818299, NSF 9802068, NSF CDA-9726389, NSF EIA-9818299, NSF CISE/CDA-9805548, NSF IRI-9896020 and NSF IRI-9530785. We are grateful to Keith Kotay for his help in many aspects of this work.

Notes

1. Clyde measures the rotational power for compliance by searching among power levels: he tries values of rotational power, until he finds the largest value that doesn't cause the robot to freely rotate.

2. All the timeouts can be replaced by sensing operations.

3. Our robots can be in one of three states: *active*, *passive*, and *dormant*. A passive robot may be in motion (for example, it may be towed or moved by an active robot). A dormant robot is stationary, and it exerts no forces directly on the manipulandum. A dormant robot may act as a fixture with respect to the rope.

References

J. Baillieul, R. Brockett, B. R. Donald, et al. *Robotics*, volume 41 of *Symposia in Applied Mathematics*. American Mathematical Society Press, Providence, RI, 1990.

J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning the the presence of obstacles. *Algorithmica*, 10(2):121–155, August 1993. Special Issue on Computational Robotics.

K.-F. Böhringer, B. R. Donald, and N. C. MacDonald. Programmable vector fields for distributed manipulation, with applications to MEMS actuator arrays and vibratory parts feeders. *IJRR* 18(2):168–200, February 1999.

K.-F. Böhringer, B. R. Donald, N. C. MacDonald, G. T. A. Kovacs, and J. W. Suh. Computational methods for design and control of MEMS micromanipulator arrays. *Computer Science and Engineering*, pages 17–29, January – March 1997.

K.-F. Böhringer, J. W. Suh, B. R. Donald, and G. T. A. Kovacs. Vector fields for task-level distributed manipulation: Experiments with organic micro actuator arrays. pages 1779–1786, Albuquerque, New Mexico, Apr. 1997.

K. Böhringer, R. Brown, B. Donald, J. Jennings, and D. Rus. Information Invariants for distributed manipulation: experiments in minimalism *Experimental Robotics IV*, Lecture Notes in Control and Information Sciences 223, eds. O. Khatib and K. Salisbury, pp. 11-25, Springer-Verlag, 1997.

B. R. Donald. *Error Detection and Recovery in Robotics*, volume 336 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 1989.

B. Donald, J, Jennings, and D. Rus. Minimalism + Distribution = Supermodularity. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(1997) 293-321, 1997.

B. Donald, J. Jennings, and D. Rus. Information invariants for cooperating autonomous mobile robots. *International Journal of Robotics Research*, 16(5) 673-702, 1997.

B. Donald, J. Jennings, and D. Rus. Information invariants for distributed manipulation. *The First Workshop on the Algorithmic Foundations of Robotics, eds. K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson*, pages 431–459, 1994.

K. Goldberg, J.C.-Latombe, D. Halperin, and R. Wilson, editors. *The Algorithmic Foundations of Robotics: First Workshop.* A. K. Peters, Boston, MA, 1995.

K. Y. Goldberg. Orienting polygonal parts without sensing. *Algorithmica*, 10(2/3/4):201–225, August/September/October 1993.

J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Press, 1991.

J.-C. Latombe. Robot algorithms. In T. Kanade and R. Paul, editors, *Robotics Research: The Sixth International Symposium*, 1993.

J. Ostrowski and J. Burdick. Geometric Perspectives on the Mechanics and Control of Robotic Locomotion. *Proceedings of the 1995 International Symposium on Robotics Research.*

J. Rees and B. Donald. Program Mobile Robots in SCHEME. *Proceedings of the 1992 International Conference on Robotics and Automation.* Nice, France 1992.

D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. *Proceedings of the 1995 Conference on Intelligent Robot Systems*. 1995.