Flexible Part Orienting Using Rotation Direction and Force Measurements

Shawn Rusaw[*], Kamal Gupta[†], Shahram Payandeh[†]

March 18, 2001

Abstract

This paper presents a novel sensor-based flexible part orienting system based around the commonly available force/torque sensor. The system orients planar parts arriving on a conveyor belt via a sequence of pushing operations with a force/torque sensor equipped fence. A method to use the raw force data from the sensor to infer the rotation direction of the part is presented. Algorithms utilising (i) only rotation direction and (ii) rotation direction plus force information are presented. These algorithms are shown to find orienting plans with fewer steps than current sensorless orienting techniques, and for a number of specified part shape classes, current sensor-based techniques. Plans generated by our algorithms were tested and verified using a conveyor/robotic car testbed.

1   Introduction

A manufacturing process is typically comprised of several stages (i) taking bulk parts from unknown orientation to known orientation (ii) transferring oriented parts to assembly site, keeping known orientation and (iii) assembling parts into completed item. Stage (i) is often done with a vibratory bowl feeder. Bowl feeders are a widely accepted method to orient parts, however they must be manually redesigned for each new part, a process which requires a certain degree of experience, if even possible [BPM82].

A long term goal of the part orienting research community is the design of a *flexible part feeder* that can orient a large variety of parts without redesign for each new part. The flexible feeder will take a computer model of the part and automatically reconfigure based on a knowledge of the part mechanics and the part's interaction with its environment. Until recently, part orienting research has focused on orienting techniques that do not require sensors (see Section 2.2). The main justifications for this being (i) current techniques like bowl feeders are sensorless and (ii) sensorless apparatus are simpler since they do not have the added complexity introduced by sensors, e.g. additional hardware and processing required.

One sensor-based commercial system, the Adept FlexFeeder[1] , has been developed for the task of part feeding. The FlexFeeder consists of a conveyor belt, machine vision system and a robotic manipulator. The vision system is used to determine the orientation of a part travelling on the conveyor and the manipulator is

---

[*]Shawn Rusaw is with the Programming Research Group, University of Oxford Computing Lab, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom. E-mail: shawn.rusaw@comlab.ox.ac.uk. This paper is based on his graduate work at Simon Fraser University.

[†]Kamal Gupta and Shahram Payandeh are with the Robotics Laboratory, School of Engineering Science, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada. E-mail: kamal@cs.sfu.ca, shahram@cs.sfu.ca.

[1]Information about the FlexFeeder can be found on Adept Technology's web page http://www.adept.com/

used to orient the part based on this information. In some manufacturing environments, vision based feeders may not be suitable. In particular, if the camera view is somehow obstructed by additional machinery then another choice of sensor is prudent.

Taylor, Mason and Goldberg [TMG87] consider the problem of sensor based part orienting as a "game with nature", and present a framework that treats manipulation planning as a tree search in a task graph. Each node in the graph is a set of possible world *states* (stable orientations of the part) and each action and sensory event causes a transition from a set of possible initial world states to a set of possible resultant world states. The graphs has an AND/OR structure because the planner can guarantee a successful plan if *any* of its actions leads to the goal state(s), however it must ensure that *all* sensory outcomes are accounted for. The planner searches the graph to reach the goal state. The resulting path to the goal state in the search tree represents a plan, i.e. an orienting strategy that will orient a part "no matter what nature decides will be the outcomes of its actions." Using this framework, Akella and Mason [AM99] used a simple diameter sensor along with a sequence of pushing operations during the orientation process to reduce the number of manipulation steps required over the sensorless case.

In this paper, we use the commonly available 6-axis force/torque sensor along with a sequence of pushing operations as the cornerstone of the first fully force-based orienting system. Firstly, we abstract the force/torque sensor into a rotation sensor by presenting a robust (despite noisy force measurements) method to determine the rotation direction (clockwise, denoted by $\circlearrowright$, or counterclockwise, denoted by $\circlearrowleft$) during orientation. We present two algorithms, one that finds optimal plans in exponential time and another that finds plans in polynomial time. The first (optimal) planner is shown to reduce the number of orientation steps compared to the current optimal sensorless and sensor-based planners. The second planner runs in polynomial time as a function of the number of stable edges in the part, and appears to be the first sensor-based polynomial time algorithm. Secondly, we use the limit surface [GRP91] to model the frictional forces between the part and conveyor, and use this to determine the *ideal forces* we expect to be applied by the part on the fence (interchangeably called a "pusher") during orientation. Our orienter then compares forces measured during orientation to these ideal forces and further reduces the number of orientation steps. Experimental results verify the utility of these techniques.

Our planning framework is modelled after that of [TMG87, AM99] in that it uses an AND/OR search procedure and a sensor, but it is novel in several ways. First and foremost is the usage of a force/torque sensor as the basis for a complete orienting system, although our results using rotation direction could be used irrespective of what type of physical sensor (say tactile strip) is used to sense it. We present a robust method to measure rotation direction using this force/torque sensor, in effect abstracting the force/torque sensor into a rotation sensor. A key aspect in our work is that the rotation direction is a fundamentally different type of information (than for example diameter) from a planning point of view. Sensors used in part orienting systems in previous works [TMG87, AM99] provide direct, but partial, information about the state of the part during the orientation process—in an abstract sense these are 'state sensors'. Formally speaking, let X denote the set of possible states and D denote the set of sensor readings. The 'state sensor' function (or model) assumed in earlier works is of the form $s_D : X \to D$, meaning the sensor provides a direct measurement of some characteristic of the state. Two or more states can be distinguished using sensor data if s_D^{-1} is different (or non-overlapping intervals, if noise is taken into account) for these states. Branching in the diameter sensor-based planner in [AM99] is based on this notion of state distinguishability, or those stable states that can be distinguished using sensor data. Rotation direction, on the other hand, does not provide direct information about the state of the part, since all states can come to rest via clockwise or counterclockwise rotation. State uncertainty is instead reduced by inferring which states could (resp. could not) lead to clockwise or counterclockwise sensor readings during transition from a given set of initial states to final states; the sensor provides partial information about 'state transition'—it is a 'state transition' sensor. Consequently, many of the concepts introduced in earlier works need to be generalised to deal with this class of 'state transition sensors'. Formally speaking, the rotation sensor function is $s_R : X \times X \to D$, with $D = \{\circlearrowright, \circlearrowleft\}$, so given an initial state, x_i and a final state x_f,

$s_R(x_i, x_f) \in \{\circlearrowright, \circlearrowleft\}$. In order to distinguish states on the basis of sensor data, we need to infer that given $x_f \in X$ *and* $d \in D$, the initial state must be x_i. In other words, we need to reason on the combined $X \times D$ space. This is the fundamental difference that arises when using a 'state transition' sensor. Correspondingly, concepts such as the resting and action ranges used in [AM99] need to be generalised by augmenting them with associated sensor readings. This leads to the new concept of *sensor augmented state distinguishability* and the notions of the *sensor resting ranges* and *sensor action ranges*, generalisations of the resting and action ranges introduced in [AM99].

We show a key property of the force/torque sensor: it allows a planner to reduces state uncertainty by at least one with each push-align operation, a characteristic lacking from the diameter sensor based planner. The worst case plan lengths (over specified subsets of the class of polygonal parts) determined by our approach are better than that with a diameter sensor (see Section 6.6). In particular, when using rotation direction, our approach results in shorter (worst case) plans when all states result in identical diameter readings. When using rotation and force, our method results in shorter plans when the push-diameter function [AM99] is symmetric or quasi-symmetric.

1.1 Overview of Our Sensor-Based Part Orienting Algorithm

Our apparatus consists of a conveyor belt, a flat fence attached to a force/torque sensor which is placed across the conveyor belt and a simple robotic manipulator. Parts are oriented through a sequence of *push-align operations*, with a single push-align operation being shown in Figure 1. Each push-align operation is followed by a sensor reading which determines which branch of a part-specific, pre-determined plan to follow. We explain the precise mechanics of these measurements in Section 7. Figure 2(a) shows an example plan using rotation direction. A node (ovals) corresponds to the set of *stable states* (stable orientations) that the part may be in at that stage, and the push-align operation corresponds to a link. Starting from the set of all unknown orientations, the part is first pushed into one of three possible stable states. If the manipulator performs a push-align operation corresponding to a rotation by $\theta_{action} = 63°$, the subsequent sensor readings reduce the number of uncertain stable states by at least one. A clockwise sensor measurement indicates the part is oriented, and a counterclockwise measurement indicates a further push-align operation through $\theta_{action} = 63°$ is required to orient the part. Figure 2(b) shows a plan using rotation direction and force to further reduce the number of steps required to orient the part.

We now explain how our algorithm derives a plan, given the part geometry and location of its centre of mass. Consider a part contacting the fence with initial angle θ as shown in Figure 1(c). Mason's Voting Theorem [Mas86] dictates the rotation direction of the part and thus specifies which state the part will come to rest in. As mentioned earlier, measuring the rotation direction gives us partial information about how the part makes a transition to a stable state. We use the notion of *distinguishable sensor augmented states* to differentiate between possible sensor augmented states (as opposed to the notion of distinguishable states used in [AM99]) to reduce state uncertainty during the orientation process. A *sensor augmented state* refers to a state *and* a sensor reading corresponding to transition to that state.

We start by presenting the notion of the *sensor resting ranges* of a part. The sensor resting ranges are those intervals of initial orientations θ that result in the same final state and sensor reading—a sensor resting range represents an *equivalence class* of initial orientations with respect to final state and sensor reading. Figure 3 gives an example of a sensor resting range diagram. The sensor resting ranges are then used to construct the *sensor action ranges* of a part, the range of actions θ_{action} (see Figure 1) that take an initial stable state to another stable state and sensor reading—an equivalence class of actions. Our optimal length planner operates by exhaustively searching the result of every possible sensor action range on every possible set of stable states, and is able to return the shortest orienting plan, if one exists, and exit with a failure if no solution exists. Our resulting plans have worst case length of n when using rotation direction or $m + 1$ when using rotation direction and force together (n refers to the number of stable states of the part and m is the size of the largest
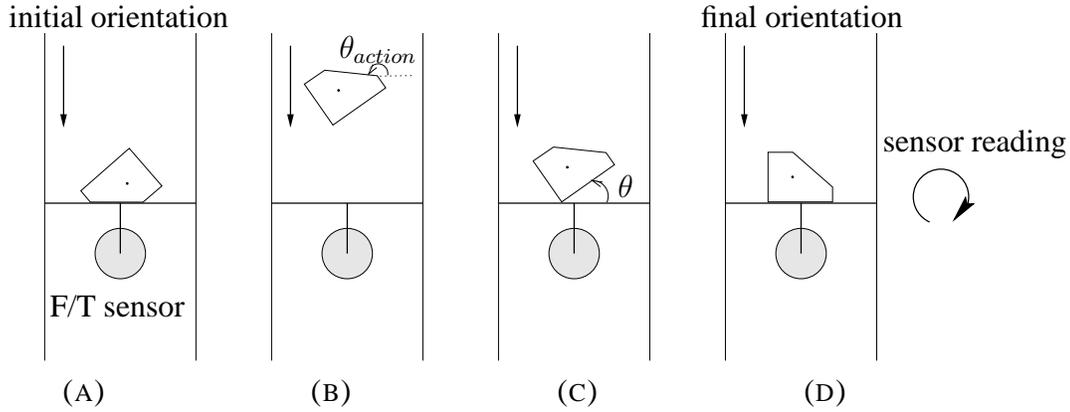
Figure 1: THE PUSH-ALIGN OPERATION WHICH CORRESPONDS TO A LINK OF THE SEARCH TREE IN FIGURE 2. (A) PART STARTS IN SOME INITIAL STABLE ORIENTATION (NODE OF SEARCH TREE) (B) PART IS PLACED UPSTREAM AFTER BEING ROTATED COUNTERCLOCKWISE THROUGH SOME ANGLE θ_{action} (θ_{action} IS THE CONTROL PARAMETER OF OUR ALGORITHM), (C) THE PART MOVES TOWARD THE FENCE AND BEGINS TO ROTATE ONCE IT CONTACTS THE FENCE (DIRECTION GIVEN BY MASON'S VOTING THEOREM) AND (D) THE PART ROTATES TO A STABLE STATE AND A SENSOR MEASUREMENT CORRESPONDING TO A STATE TRANSITION IS MADE.

indistinguishable set of sensor augmented states).

Force information is then incorporated to further enhance the performance of our planner. We use the limit surface model of quasi-static pushing to determine the *ideal force* $\mathbf{f}_{ideal}$ we expect to measure just before the part comes to rest for each clockwise/counterclockwise state transition. Transition to a particular stable state can be made via clockwise or counterclockwise rotation, each leading to possibly different forces between the part and fence, so force is also a state transition dependent quantity. Figure 4 shows that a part may have different ideal forces for clockwise ($\mathbf{f}_{ideal}^{\circlearrowright}$) and counterclockwise ($\mathbf{f}_{ideal}^{\circlearrowleft}$) rotation, depending on the distance from the vertex to the centre of mass. Forces measured during orientation $\mathbf{f}_m$ can then be compared to these ideal forces to further distinguish sensor augmented states, leading to worst case plans with fewer steps. It should be noted that issues of sensor noise and model inaccuracies may limit the usefulness of force data to lopsided parts, i.e. parts with disparate distances from each vertex to the centre of mass, a hammer for example.

2   Related Work

2.1   Pushing as a Orientation Technique

Using quasi-static pushing as a manipulation primitive was first examined by Mason [Mas86]. Mason showed that pushing occurs in the early stages of grasping a part and is useful in reducing uncertainty in a part's orientation. He developed a rule to determine the rotation direction of a part, based only on the location of the centre of mass of the part. This last result forms the basis for a great deal of work in automated assembly as it gave a way to determine the gross behaviour (rotation direction) of a pushed part.

Peshkin and Sanderson [PS88a], determined bounds on the rotation rate of a part, making no assumptions about pressure distributions between the part and the supporting surface. This work made it possible to determine the slowest/fastest rotation rate and velocity possible for a given contact configuration. Alexander and Maddocks [AM93] also determined bounds on the possible motions of a pushed part, but their result did not use the conservative circular approximation in [PS88a]. Goyal, Ruina and Papadopoulos [GRP91] determined the relationship between the net contact force and resultant velocity of a part—the Limit Surface—when the
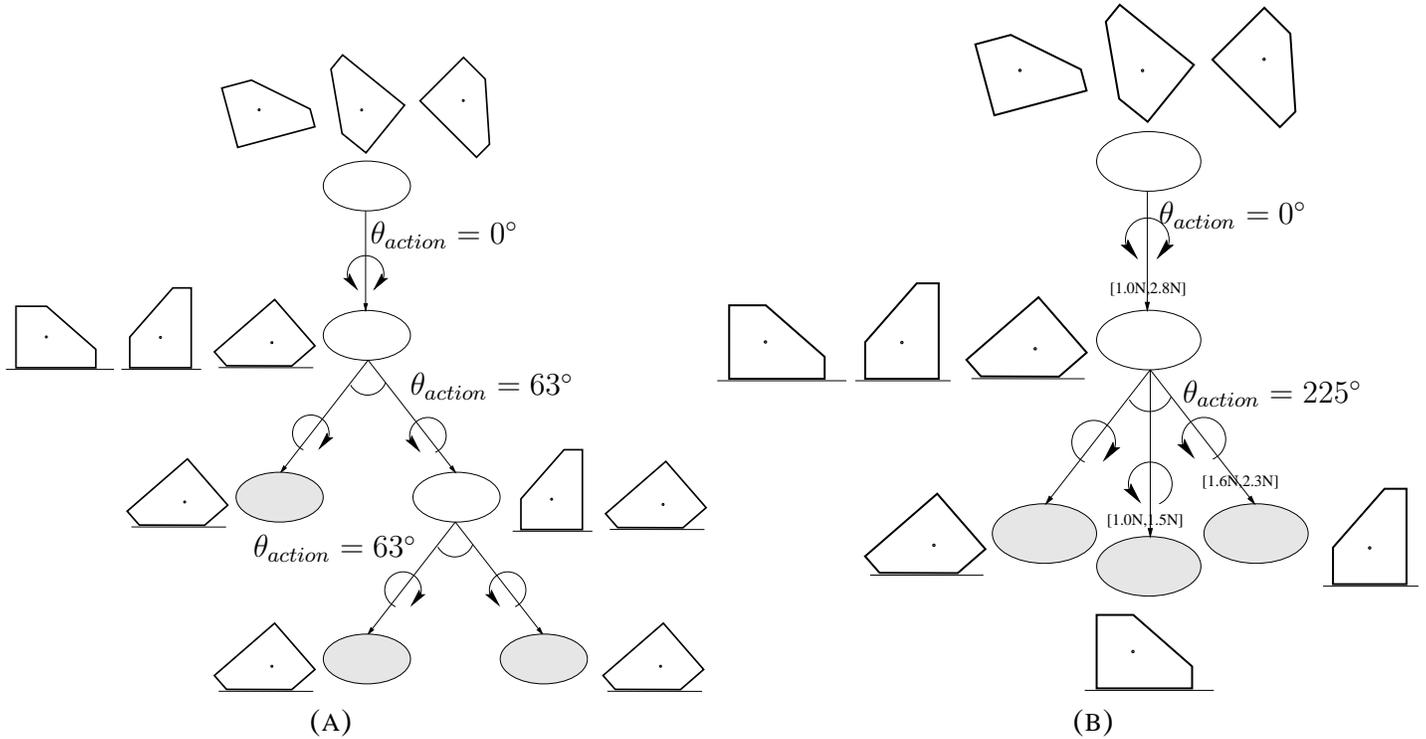
(A)　　　　　　　　　　　(B)

Figure 2: EXAMPLES OF ORIENTING PLANS. (A) SHOWS A PLAN USING ONLY ROTATION DIRECTION AND (B) GIVES A PLAN USING ROTATION DIRECTION AND FORCE. IN (A), AFTER THE FIRST SENSOR READING, ROTATION DIRECTION PROVIDES NO INFORMATION ABOUT THE STATE OF THE PART. HOWEVER, EACH LEVEL OF THE PLAN REDUCES THE UNCERTAINTY BY ONE STATE, BASED ON THE SENSOR READING. IN (B) THE ADDITION OF FORCE INFORMATION FURTHER REDUCES THE NUMBER OF PUSH-ALIGN OPERA-TIONS REQUIRED. AFTER THE FIRST PUSH-ALIGN OPERATION, THE SENSOR DATA IS INDISTINGUISHABLE FOR ALL STATES. HOWEVER, THE SENSOR DATA AFTER A ROTATION OF $\theta_{action} = 225°$ IS DISTINGUISH-ABLE, LEADING TO A FULLY ORIENTED PART. A CLOCKWISE SENSOR READING INDICATES A SINGLE STATE AND STATES WITH COUNTERCLOCKWISE SENSOR READINGS CAN BE FURTHER DISTINGUISHED BY THE FORCE.
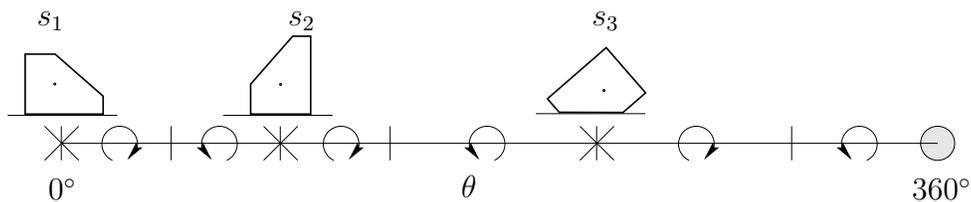
PSfrag replacements

Figure 3: SENSOR RESTING RANGE DIAGRAM. s_1, s_2 AND s_3 REPRESENT STABLE STATES OF THE PART. THE $\times$'S INDICATE STABLE ORIENTATIONS, THE VERTICAL BARS $|$ SPECIFY THE LIMITS OF THE SENSOR RESTING RANGE AND THE ⟳ OR ⟲ REPRESENT THE SENSOR READING AS THE PART MAKE A TRANSITION TO THE STABLE STATE.
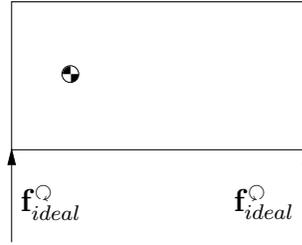
PSfrag replacements

$\mathbf{f}_{ideal}^{\circlearrowright}$ $\mathbf{f}_{ideal}^{\circlearrowleft}$

Figure 4: THE LIMIT SURFACE DICTATES THAT $\parallel \mathbf{f}_{ideal}^{\circlearrowright} \parallel > \parallel \mathbf{f}_{ideal}^{\circlearrowleft} \parallel$, DUE TO THE DIFFERENT DISTANCES BETWEEN THE VERTICES AND THE CENTRE OF MASS.

pressure distribution is assumed known. This has been used to localise the position of a part using a closed loop system by Lynch, Maekawa and Tanie [LMT92].

Using pushing in the parts transfer stage—stage (ii) of the manufacturing process outlined in the Introduction—has also been studied. Lynch [Lyn92] studied the motion of a part being pushed along by its edge with a flat pusher. He determined the range of *stable* pushing directions where the part does not rotate to another edge. Lynch and Mason [LM96] examined the controllability of pushing and developed a planner to find push sequences to move a part about in an obstacle cluttered workspace. Akella [Ake96] developed a planner that can take a part from any known initial configuration to any final specified goal position and orientation using a sensorless sequence of pushes.

2.2 Orienting Without Sensors

Sensorless orientation of parts has been examined by Mason [Mas86] who used pushing to reduce the uncertainty in a part's orientation. Mani and Wilson [MW85] used Mason's result to develop a system that orients parts with a sequence of fence pushes at different angles. Peshkin and Sanderson [PS88b] developed a planner for orienting parts with fences suspended above a conveyor belt. This planner used conservative bounds on the possible motions of a workpiece to plan the angles of a sequence of fences above a conveyor belt. As the conveyor pushes the parts past each subsequent fence, the uncertainty in part orientation is reduced, and after the final fence the orientation of the part is known within a small angle range. Brokowski, Peshkin and Goldberg [BPG95] improved on this idea by designing curved fences which further reduce the uncertainty in part orientation to a specific single orientation. Recently, a complete algorithm for planning $O(n^2)$ length curved fence assemblies has been presented by Berretty, Goldberg, Overmars and van der Stappen [BGOvdS98] for the frictionless case.

Brost [Bro88] showed how to plan parallel jaw grasping motions to orient a part in the presence of uncertainty. Using Mason's results he was able to show the behaviour (final grasped orientation) of a part being grasped by a parallel-jaw gripper. Goldberg [Gol93] showed that a part can only be oriented to symmetry in its push function by a sensorless planner, and developed the complete *backchaining algorithm* to plan a series of parallel jaw motions to orient a part to symmetry. Using a frictionless assumption and the development of the *diameter function*, he was able to determine in $O(n^2)$ time, a sequence of parallel jaw grasps to orient a part to symmetry in $O(n^2)$ steps. It has since been shown by Chen and Ierardi [CI92] that the backchaining algorithm can in fact generate $O(n)$ length plans with a worst case of $2n - 1$. Akella, Huang, Lynch and Mason [AHLM97] have developed a system that orients a part on a conveyor using a single fence with 1 DOF. This technique implements the backchaining algorithm with a single jointed fence above a conveyor belt

A vibratory system developed by Sony—Sony APOS system [SS89]—orients parts utilising a vibrating tray with 'cups' that accept parts that are in the correct orientation. This system is closely related to the methodology of the vibratory bowl feeder.

2.3 Orienting With Sensors

Lynch, Maekawa and Tanie [LMT92] utilised the limit surface model [GRP91] of quasi-static motion to develop a closed loop system that localised the position of a part. They used a round robotic fingertip equipped with a tactile array to push a part from a bounded unknown orientation to a known orientation. Jia and Erdmann [JE96] developed a system to determine the orientation of a part by pushing with a round, tactile sensor equipped fingertip. By observing the motion of the contact point as the part is pushed, they are able to determine the final position of the part. Salvarinov and Payandeh [SP97] utilise a single joint, strain gauge equipped fence above a conveyor and detect the contact signature of a part interacting with the fence. The contact signature is used to determine orientation and orient the part. Rao and Goldberg [RG94] characterised orientability and recognisability of multiple parts by parallel jaw grasping and diameter sensing.

Our paper is most closely related to the research of Akella and Mason [AM99] who showed that utilising diameter sensor data during the orientation process reduces the number of manipulation steps required over the sensorless case. Two key contributions of [AM99] were (i) parts with symmetric push functions can be *fully* oriented if the push-diameter function is asymmetric and (ii) multiple similarly shaped parts can have the same push or push-diamter functions, and thus orienting plans can be found to orient multiple parts. We discuss our work in the context of (i) and (ii) in Section 6.6.

3   Assumptions

The orienting techniques outlined in the remainder of this paper require the following assumptions:

1. Parts are polyhedra which have constant polygonal cross-sections. Non-convex cross-sections are considered by using their convex hull.
2. The location of the part vertices and centre of mass are known.
3. Motion is quasi-static (conveyor moves slowly).
4. Coefficient of friction μ is constant over sliding support surface.
5. Frictional interactions are described by Coulomb model.
6. All pushes are perpendicular to the conveyor motion.
7. The parts and fence are perfectly rigid.

Two additional assumptions are required to determine the ideal forces using the limit surface model, which requires a known applied force direction and a known pressure distribution between the part and the sliding surface:

8. There is zero friction between part and fence. We then know the forces are perpendicular to the fence.
9. We use a constant, uniform distribution to find the ideal forces, but account for variations in Section 7.2.2.

All angle arithmetic is done $\mod 2\pi$, and all index arithmetic is done $\mod n$, where n is the number of stable states of the part.

4   Definitions

This section presents several definitions required for the development of our work. Section 4.1 introduces some background definitions and Section 4.2 introduces notions relating to sensor augmented state distinguishability. Sections 4.3 through 4.5 introduce the concepts of sensor resting and action ranges using only rotation direction data and Section 4.6 discusses incorporating force data with rotation direction.
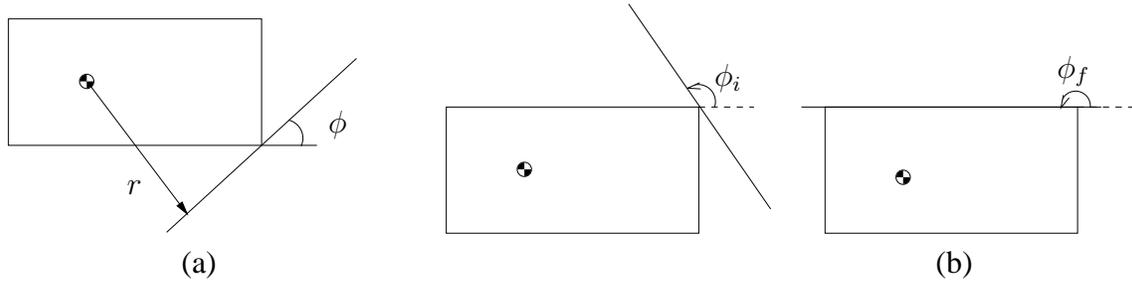
Figure 5: (A) THE RADIUS FUNCTION IS THE DISTANCE FROM THE CENTRE OF MASS TO THE SUPPORT LINE r AS A FUNCTION OF THE SUPPORT LINE ORIENTATION ϕ. (B) NOTATION USED IN DESCRIBING THE PUSH FUNCTION: ϕ_i IS ORIENTATION OF THE SUPPORT LINE AT BEGINNING OF LINEAR NORMAL PUSH AND ϕ_f IS ORIENTATION AT THE END.

4.1   The Radius, Push and Push-Sensor Functions

The *radius function* was first used for part orienting by Goldberg [Gol93] in the context of orienting a polygonal shaped part through a series of parallel jaw grasps. Figure 5(a) shows the construction method of the radius function. The support line is considered to be a flat pusher (the fence in our technique), ϕ is the angle of orientation of the support line with respect to the part and r is the perpendicular distance from the centre of mass to the support line.

Definition 4.1 The **radius function** of a polygon is a mapping $f_r : S^1 \to \mathbb{R}$ from the orientation ϕ of the support line of a polygon to the perpendicular distance r from the polygon's centre of mass to the support line. ☐

A part being pushed in a direction perpendicular to the fence (a *linear normal push*) tends to rotate towards local minima in the radius function. So, given the part orientation and push direction, the radius function allows us to determine the stable state and rotation direction of the part. This information is represented by the push and push-sensor functions. In the following definitions ϕ_i is the orientation of the support line at the start of the push and ϕ_f is the orientation at the end (see Figure 5(b)). Note that rather than a stationary fence and rotating part as is the case in our physical set-up, conceptually it is assumed that the part is stationary and the pusher rotates (of course in opposite direction but same magnitude).

Definition 4.2 The **push function** of a part is a mapping $f_p : S^1 \to S^1$ from the initial fence orientation ϕ_i to the final fence orientation ϕ_f of a linear normal push. ☐
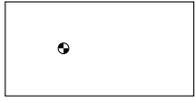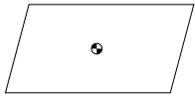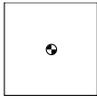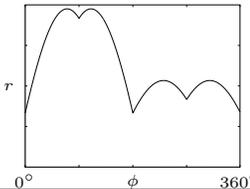
It is known that a part can only be oriented to symmetry in its push function by a sensorless planner [Gol93]. A symmetric push function has a 'period' of $T < 2\pi$ such that $f_p(\phi + T) = (f_p(\phi) + T) \bmod 2\pi$ (note that periodicity here is along the ordinate axis, and not along the abscissa, the usual definition of a periodic function).

Two stable states s_i and s_j corresponding to final fence orientations ϕ_f^i and ϕ_f^j are termed *equivalent* with respect to a symmetric push function if $|\phi_f^i - \phi_f^j| \bmod T = 0$. We use $s_i \doteq s_j$ to denote this. We define a *set S_e of equivalent states* as a set with all members equivalent (i.e. for all $s_i, s_j \in S_e$, $s_i \doteq s_j$) and define a *set S_n of non-equivalent states* as a set containing no two states $s_i \doteq s_j$.
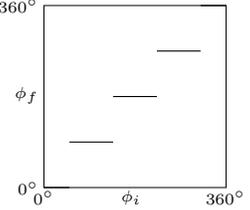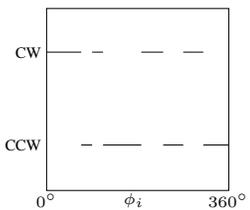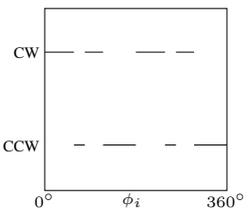
Definition 4.3 The **push-sensor function** of a part is a mapping $f_{ps} : S^1 \to D$ that takes the initial fence orientation ϕ_i to the domain D of the sensor at the end of a linear normal push. ☐

Definition 4.4 A part has a **symmetric** push-sensor function if there exists a period T, $0 < T < 2\pi$, such that $f_{ps}(\phi) = f_{ps}(\phi + T)$ and $f_p(\phi_f + T) = \phi_f + T \bmod 2\pi$ for every stable orientation ϕ_f. ☐

Table 1: SOME EXAMPLES OF THE PERIOD OF THE RADIUS, PUSH AND PUSH-SENSOR (ROTATION DI-RECTION) FUNCTIONS. p IS THE NUMBER OF PERIODS IN THE PUSH FUNCTION, AND INDICATES THE SYMMETRY OF THE PART.

Parts (in orientation $\theta = 0°$)			
Radius function (not to scale)			
Push function			
Push-sensor function			
# of periods - p	1	2	4
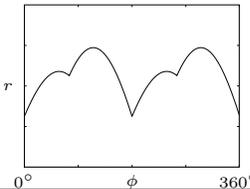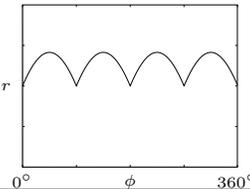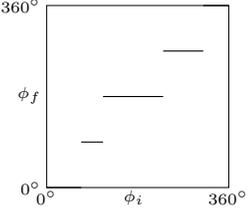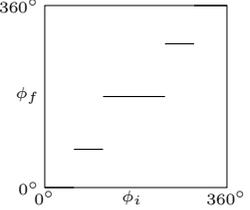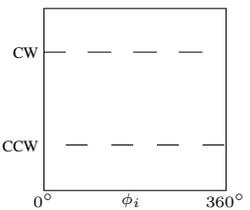
We use these defintion primarily in Section 6.6 when comparing our results to those in [AM99]. Table 1 contains three examples of the radius, push and push-rotation direction functions for parts with various levels of symmetry in the push and push-sensor functions.

4.2 Sensor and Sensor Augmented State Distinguishability

Figure 6 briefly explains (details later in Section 7) the key idea behind measuring rotation direction. As the part rotates into alignment with the fence, it will exert a force $\mathbf{f}_m$ and moment $\mathbf{m}_m$ on the fence. Balancing the force and moment results in a derived contact location at vertex of the part. As the part rotates onto a stable edge, this derived contact location will jump to a location below the centre of mass. Detecting the direction of this jump, which can be very reliably extracted even from noisy force data, gives the rotation direction.

The force $\mathbf{f}_m$ used for orienting purposes is that measured *just before* the jump in contact location occurs. We can then assume the edge is horizontal and the force is applied to a vertex, the same assumption as when determining the ideal forces $\mathbf{f}_{ideal}$ to which $\mathbf{f}_m$ is compared during orientation. Forces measured after the step are not unique to sensor augmented states and are in fact all equal to the product of the weight of the part mg and the coefficient of friction μ. We now introduce the notion of sensor augmented state distinguishability

Definition 4.5 An **indistinguishable set of sensor readings** is a set of sensor readings (taken from the domain of possible sensor readings) where each reading is indistinguishable from at least one other reading. □

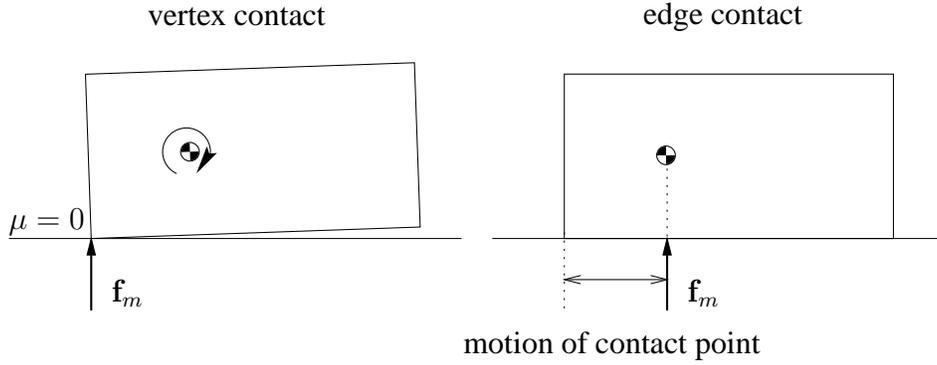vertex contact   edge contact

motion of contact point

Figure 6: ROTATION DIRECTION IS DETERMINED FROM THE MOTION OF THE CONTACT POINT DERIVED FROM THE MEASURED FORCE $\mathbf{f}_m$ AND MOMENT $\mathbf{m}_m$ (NOT SHOWN).
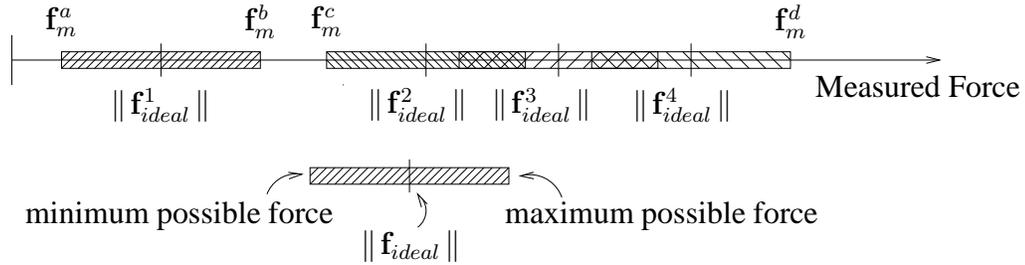

Figure 7: DISTINGUISHABILITY OF FORCE MEASUREMENTS FOR CLOCKWISE ROTATION (A SIMILAR FIGURE EXISTS FOR COUNTERCLOCKWISE ROTATION). THE FORCE VALUE THAT MATCHES THE LIMIT SURFACE MODEL ($\| \mathbf{f}_{ideal}^i \|$) FOR EACH STABLE STATE s_i IS HIGHLIGHTED BY A VERTICAL LINE. THE RANGE OF FORCES ARISES FROM SENSOR NOISE, UNCERTAIN PHYSICAL CHARACTERISTICS AND DEVIATIONS FROM MODEL-BASED BEHAVIOUR.

Definition 4.6 An **indistinguishable set of sensor augmented states** is a set of indistinguishable sensor readings *and* a set of states which when transitioned to result in the indistinguishable set of sensor readings. □

The domain of rotation direction is binary $\{\circlearrowright, \circlearrowleft\}$, so when using only rotation direction, there are only two indistinguishable sets of sensor augmented states, one corresponding to clockwise $\{\{\circlearrowright\}, \{s_1, \cdots, s_n\}\}$ and the other to counterclockwise $\{\{\circlearrowleft\}, \{s_1, \cdots, s_n\}\}$. By convention, the *size* of an indistinguishable set of sensor augmented states is given by the number of states in the set. Since each of these sets contain all the stable states, they each have size=n.

On the other hand, the domain of force data is the set of non-negative reals (alternately we may bound this domain above by a force larger than can possibly be measured). Force measurements for each state and rotation direction are closed subintervals of this domain. The size of these intervals is determined from sensor noise and measurement error as outlined in Section 7.2.2. Recall from Figure 4 that a state may have different values of $\mathbf{f}_{ideal}^{\circlearrowright}$ and $\mathbf{f}_{ideal}^{\circlearrowleft}$, so when combined with rotation direction, force data acts to split the indistinguishable sets of sensor augmented states (using rotation direction alone) into smaller subsets. Figure 7 shows a set of force intervals for a hypothetical part. The figure corresponds to clockwise rotation and shows that force data splits the indistinguishable set of sensor augmented states for rotation direction alone $\{\{\circlearrowright\}, \{s_1, s_2, s_3, s_4\}\}$ (size=4) into the smaller sets $\{\{\circlearrowright, [\mathbf{f}_m^a, \mathbf{f}_m^b]\}, \{s_1\}\}$ (size=1) and $\{\{\circlearrowright, [\mathbf{f}_m^c, \mathbf{f}_m^d]\}, \{s_2, s_3, s_4\}\}$ (size=3). Notice that the sensor measurements for states s_2 and s_4 are individually distinguishable, since indistinguishability is not transitive, due to sensor noise [AM99].

For a particular part, different indistinguishable sets of sensor augmented states can share states. This is in
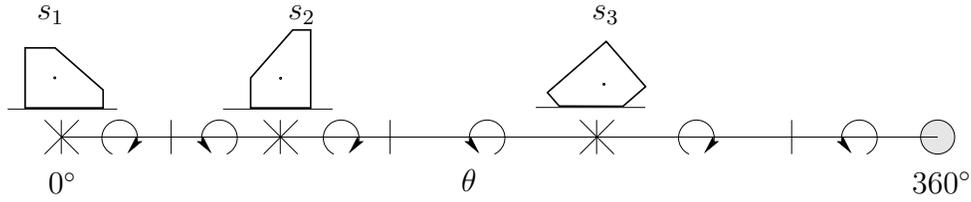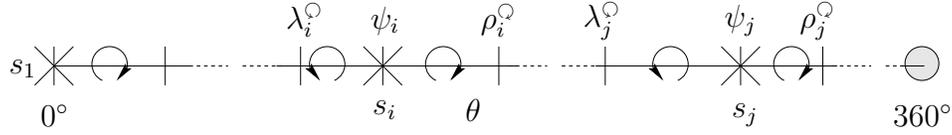
Figure 8: SENSOR RESTING RANGE DIAGRAM.

Figure 9: NOTATION FOR SENSOR RESTING RANGES. THE TWO STATES s_i AND s_j USED IN THIS EXAMPLE ARE THE SAME STATES REFERRED TO IN FIGURE 10.

contrast to the indistinguishable sets of [AM99] which will never have the same states.

4.3 Sensor Resting Ranges

The radius function can be used to determine the stable orientation ψ and rotation direction of a part contacting the pusher at an angle of θ (see Figure 1(c)).

Definition 4.7 A **sensor resting range** of a stable orientation is the set of initial orientations which lead to the same sensor reading while making a transition to that stable orientation. $\square$

The sensor resting ranges are easily derived from the location of the maxima and minima of the part's radius function. The *sensor resting range diagram* is used to represent this information. In Figure 8, s_1, s_2 and s_3 represent the stable states of the part. The stable orientation of a sensor resting range is indicated by an $\times$ and the left and right limits of the sensor resting ranges are delimited by vertical bars $|$. Each range has a $\circlearrowright$ or $\circlearrowleft$ to signify the sensor reading for that range.

Figure 9 shows the notation used to describe the sensor resting ranges. For a particular stable state s_i, the stable orientation is denoted ψ_i. Since any stable state s_i may be reached via clockwise or counterclockwise rotation, each stable state has two associated sensor resting ranges. The left and right limits of the sensor resting ranges corresponding to clockwise rotation are $\lambda_i^{\circlearrowright}$ and $\rho_i^{\circlearrowright}$. Likewise, the left and right limits of the sensor resting ranges corresponding to counterclockwise rotation are $\lambda_i^{\circlearrowleft}$ and $\rho_i^{\circlearrowleft}$. Since the stable orientation ψ_i is also a limit of its two sensor resting ranges, we have the following identities where $i+1$ (resp. $i-1$) corresponds to the state directly to the right (resp. left) on the sensor resting range diagram:

$$\text{(a)} \quad \psi_i = \lambda_i^{\circlearrowleft} = \rho_i^{\circlearrowright} \qquad \text{(b)} \quad \rho_i^{\circlearrowleft} = \lambda_{i+1}^{\circlearrowright} \qquad \text{(c)} \quad \lambda_i^{\circlearrowright} = \rho_{i-1}^{\circlearrowleft} \qquad (1)$$

As well, since the angle ϕ used in the push function $f_p(\phi)$ (resp. the angle θ of the sensor resting ranges) corresponds to clockwise rotation (resp. counterclockwise) of the part with respect to the pusher, we have the following identities valid at the stable orientations (n refers to the number of stable states).

$$\begin{aligned}
\phi_i &= 2\pi - \psi_i \\
f_p(2\pi - \psi_i) &= 2\pi - \psi_i \text{ for all } i \in \{1, \cdots, n\}
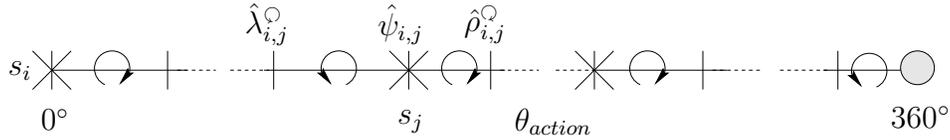\end{aligned} \qquad (2)$$

Figure 10: NOTATION FOR SENSOR ACTION RANGES. $0°$ CORRESPONDS TO MOVING THE PART UPSTREAM WITHOUT ROTATION.

4.4 Sensor Action Ranges

We now use the sensor resting ranges to develop the concept of the *sensor action range*. Recall the definition of an *action* from Figure 1 to be the counterclockwise angle θ_{action} the part is rotated through after being pushed into alignment at a previous step. It is informative to note that θ_{action} is the control variable of our subsequent search algorithms, and at each level of the search, each link of the search tree corresponds to a different possible push-align action.

Definition 4.8 A **sensor action range** is a contiguous interval of rotations that form an equivalence class of actions and sensor readings. That is, every action that belongs to a sensor action range results in the same stable state and sensor measurement for a given initial stable state. □

The sensor action ranges are constructed using the sensor resting ranges. Let the stable orientations for stable state s_i and stable state s_j be ψ_i and ψ_j respectively. Let the left and right limits of the clockwise sensor resting range for state s_j be $\lambda_j^\circlearrowright$ and $\rho_j^\circlearrowright$, respectively. A counterclockwise rotation of the part through $\theta_{action} = (\psi_j - \psi_i)$ (recalling $\mod 2\pi$ arithmetic) will cause the part to go directly from stable state s_i to stable state s_j. In fact, since the sensor resting ranges indicate a range of initial orientations that lead to the same stable state and sensor value, rotation of stable state s_i through a range of angles will lead to stable state s_j and clockwise rotation. This range is specified by the open interval $((\lambda_j^\circlearrowright - \psi_i), (\rho_j^\circlearrowright - \psi_i))$. This range is an equivalence class of actions that leads to transition from stable state s_i to stable state s_j with clockwise rotation. In a similar manner, the counterclockwise sensor action ranges are determined by the interval $((\lambda_j^\circlearrowleft - \psi_i), (\rho_j^\circlearrowleft - \psi_i))$.

Figure 10 shows the notation used to describe the sensor action ranges for transition from s_i to s_j (corresponding to Figure 9). The action corresponding to direct transition from s_i to s_j is $\hat{\psi}_{i,j}$. The left and right limits of the clockwise sensor action range are $\hat{\lambda}_{i,j}^\circlearrowright$ and $\hat{\rho}_{i,j}^\circlearrowright$ and the counterclockwise sensor action range limits are $\hat{\lambda}_{i,j}^\circlearrowleft$ and $\hat{\rho}_{i,j}^\circlearrowleft$. Since the action $\hat{\psi}_{i,j}$ is also a limit of its two sensor action ranges, we have the following identities.

$$\text{(a)} \quad \hat{\psi}_{i,j} = (\psi_j - \psi_i) = \hat{\lambda}_{i,j}^\circlearrowleft = \hat{\rho}_{i,j}^\circlearrowright \qquad \text{(b)} \quad \hat{\lambda}_{i,j}^\circlearrowright = (\lambda_j^\circlearrowright - \psi_i) \qquad \text{(c)} \quad \hat{\rho}_{i,j}^\circlearrowleft = (\rho_j^\circlearrowleft - \psi_i) \quad (3)$$

It is important to note that rotation through $(\psi_j - \psi_i)$, $(\lambda_j^\circlearrowright - \psi_i)$, and $(\rho_j^\circlearrowleft - \psi_i)$, the endpoints of the sensor action ranges, may result in several undesirable configurations. The first possible configuration occurs when a vertex of the part contacts the fence such that the force vector normal to the fence is directed through the centre of mass. The part will then not rotate until acted upon by an external force. In order to avoid this configuration, actions should be chosen from the middle of the sensor action range, thus ensuring rotation. The second configuration occurs when the part contacts the fence with a stable edge, in which case no valid state transition data is collected. This configuration is dealt with Section 5.2

The utility of the action ranges lie in their division of a continuous set of actions $[0°, 360°)$ into groups of equivalent actions—actions that lead to the same stable state and sensor reading. From a planning point of view, this reduces the search space from a continuous space (infinite branching level) to a discrete search space (finite branching level). Figure 11 shows the sensor action ranges for our example part.
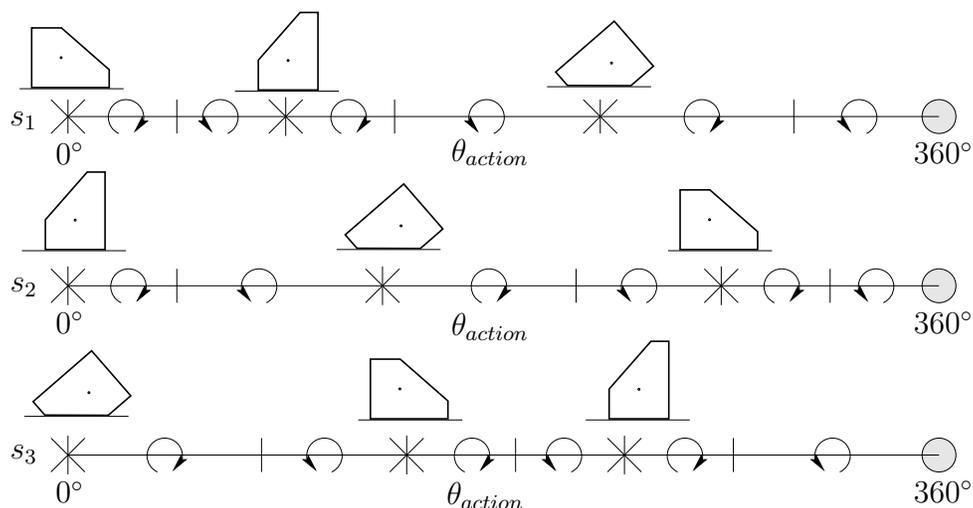
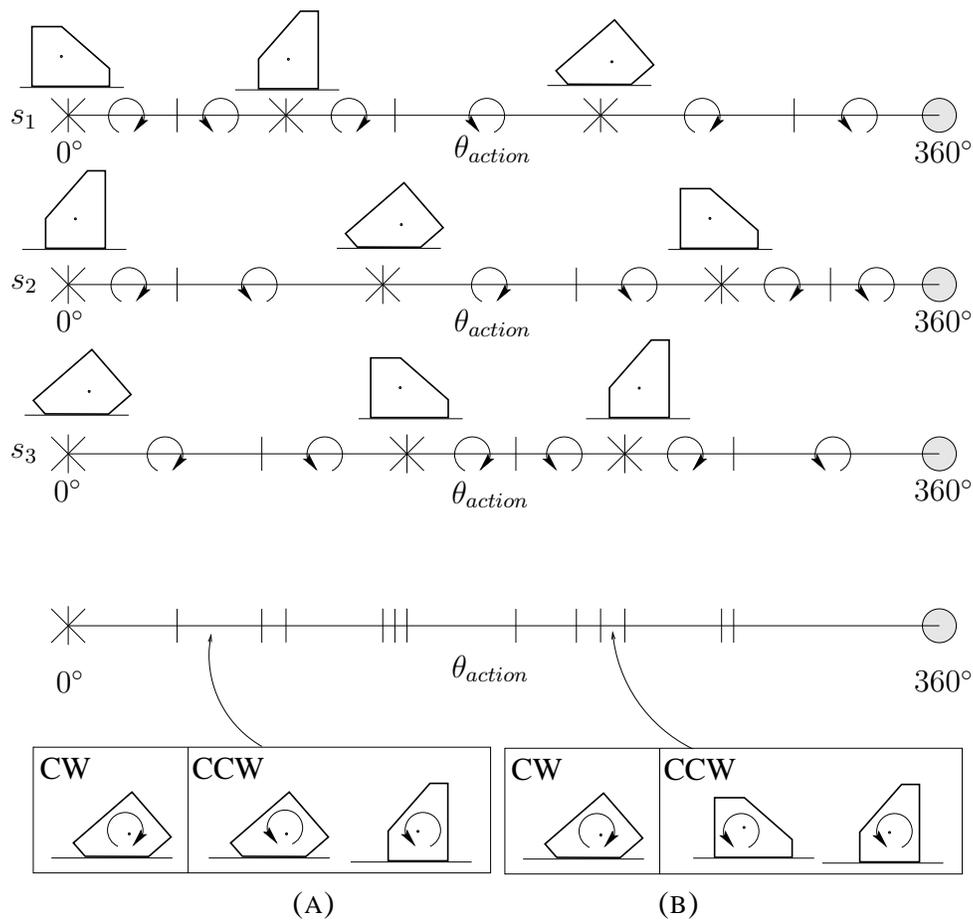Figure 11: SENSOR ACTION RANGES FOR ALL INITIAL STATES s_1, s_2 AND s_3.

Figure 12: OVERLAP RANGES WITH SENSOR INFORMATION. A SENSOR REPRESENTATIVE ACTION SE-
LECTED FROM RANGE (A) OR (B) REDUCES THE UNCERTAINTY IN PART ORIENTATION BY ONE OR TWO
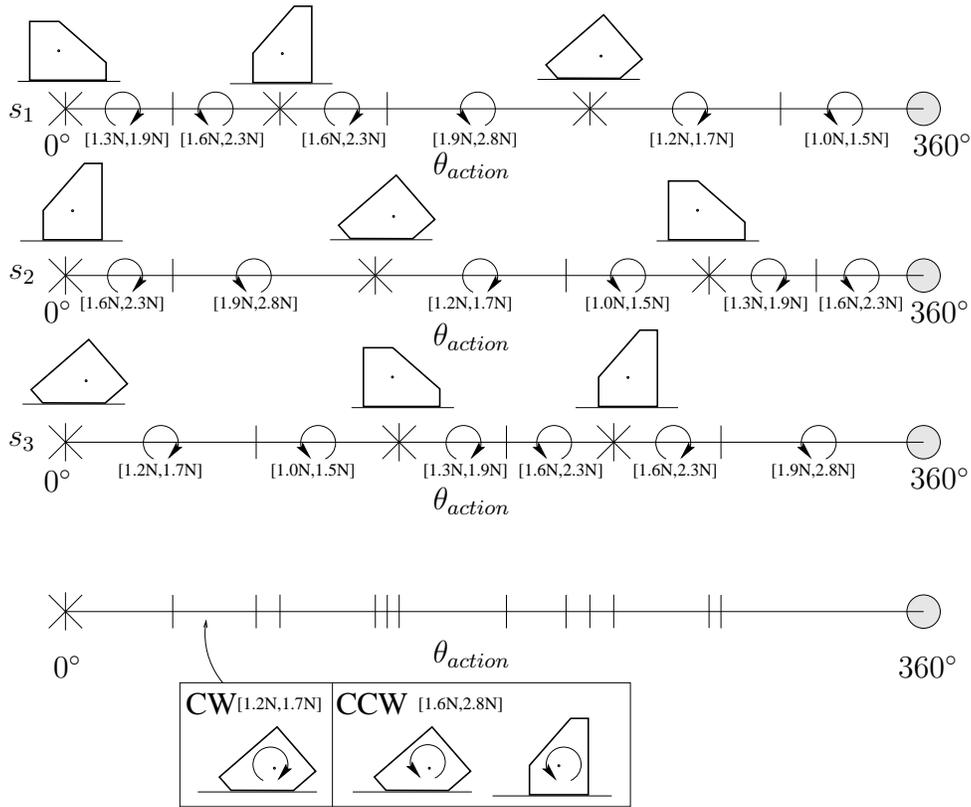STATES BASED ON CLOCKWISE OR COUNTERCLOCKWISE SENSOR READING.

Figure 13: SENSOR REPRESENTATIVE ACTIONS USING BOTH FORCE AND ROTATION DIRECTION INFORMATION. THE FORCE RANGE CALCULATIONS ARE BASED ON A PART WITH $mass = 1kg$, $\mu = 0.3$, $g = 9.81m/s^2$.

4.5 Sensor Representative Actions

Up until this point we have been discussing the effect of an action θ_{action} on a single stable state. In order to develop a planner, we need to understand the effect of an action on a group of stable states. This is done by merging the endpoints of all the sensor action ranges into a set of *overlap ranges*. Figure 12 shows the set of overlap ranges with sensor information for all stable states. The overlap ranges define an equivalence set of actions for a group of stable states in the same way as the sensor action ranges define an equivalence set of actions for a single stable state.

When the sensor information corresponds to rotation direction, we have $2kn$ sensor action ranges and a maximum of $2kn - k + 1$ overlap ranges for k stable states (in Figure 12 there are less than 16 ($n = k = 3$) overlap ranges since some of the limits of the sensor actions ranges are equal). For a given set of stable states, any action chosen from these overlapping ranges will result in the same set of stable orientations and sensor data. To stay with the tone of previous sections we will call the centre of each overlapping range a *sensor representative action*. This action is chosen to 'represent' the other actions in the range during the planning process, and is chosen from the centre of the overlap range in order to avoid the problems at the endpoints of the sensor action ranges (see Section 4.4).

4.6 Incorporating Force Data

Since a part may rotate onto a stable edge via clockwise or counterclockwise rotation, each resulting in possibly different ideal forces between the part and the fence, the addition of force data to the sensor resting and action

ranges is straightforward. The limits of the ranges do not change, but now each sensor resting/action range has a force range associated with it. The centre of this force range is specified by the limit surface model of quasi-static pushing and is calculated using the methods of Section 7.2. Figure 13 shows the sensor representative actions (force ranges indicated below each sensor action range) of our sample part when using rotation direction and force.

5 Sensor-Based Algorithms

We now present our optimal (in terms of number of steps in resultant plan) planner based on a breadth-first AND/OR search which is essentially identical to algorithms presented in [AM99, TMG87]. The time complexity to search for an optimal plan is exponential, as the search is exhaustive. A polynomial time algorithm that utilises a critical property of sensor action ranges is presented in Section 6.4. However, this polynonial time algorithm does not guarantee an optimal plan. In each case, the algorithms must be run offline once for each part, generating a conditional plan which can be followed based on the sensor data collected during the orientation process. The time complexity terms ($T_A(k, n)$, $T_B(k, n)$ etc.) are from [AM99] except when noted.

5.1 Algorithm for Optimal Length Plans

For a fine description of AND/OR search and the similar AO* algorithm, see [RK91]. Each level of the search tree corresponds to a push-align operation. The root node of the tree is the set of all possible orientations of the part. A node in the tree corresponds to the set of possible states consistent with the sequence of push-align operations and sensor readings up to that node. All links are AND links corresponding to the set of child nodes generated from the same sensor representative action θ_{action}. Each branch of an AND link corresponds to an indistinguishable set of sensor readings, and each branch + child node corresponds to an indistinguishable set of sensor augmented states.

To generate the child nodes for a base node, we must apply all the possible sensor representative actions and group the resulting sensor augmented states for each action into indistinguishable sets. At each node we generate a list of sensor representative actions for its k constituent states. This can be done in $T_A(k, n) = O(kn \log k)$ time. The time to generate all possible resultant sensor augmented states for a single action on a set of k states is $T_B(k, n) = O(k \log n)$. The time to sort these sensor augmented states into indistinguishable sets is $T_C(k, n) = O(k)$ for rotation direction and $T_C(k, n) = O(k \log k)$ otherwise. And finally, the time to remove duplicate sensor augmented states is $T_D(k, n) = O(k)$.

Starting at the root node, the first push-align operation results in the sets of indistinguishable sensor augmented states. The search generates child nodes in a breadth-first manner by applying all the sensor representative actions of the parent node's constituent states. A node is solved if it contains a single state or when all its child nodes are solved. When a node is solved, its parent node is updated. The search terminates when the root node is solved or all nodes have been explored.

5.2 Dealing with a Special Degenerate Case

If the part contacts the fence directly on a stable edge, no rotation will occur, so no valid direction or force state transition data will be available. This can only happen in the first push since subsequent sensor representative actions are chosen from the middle of the overlapping ranges, ensuring rotation (see Figure 13). When planning using only rotation direction, collecting valid state transition data in the first push is not important, since rotation direction does not reduce state uncertainty in the first step—the first nodes after the root node are always the set of all stable states. However, if no valid force data is taken, an additional push-align operation, corresponding
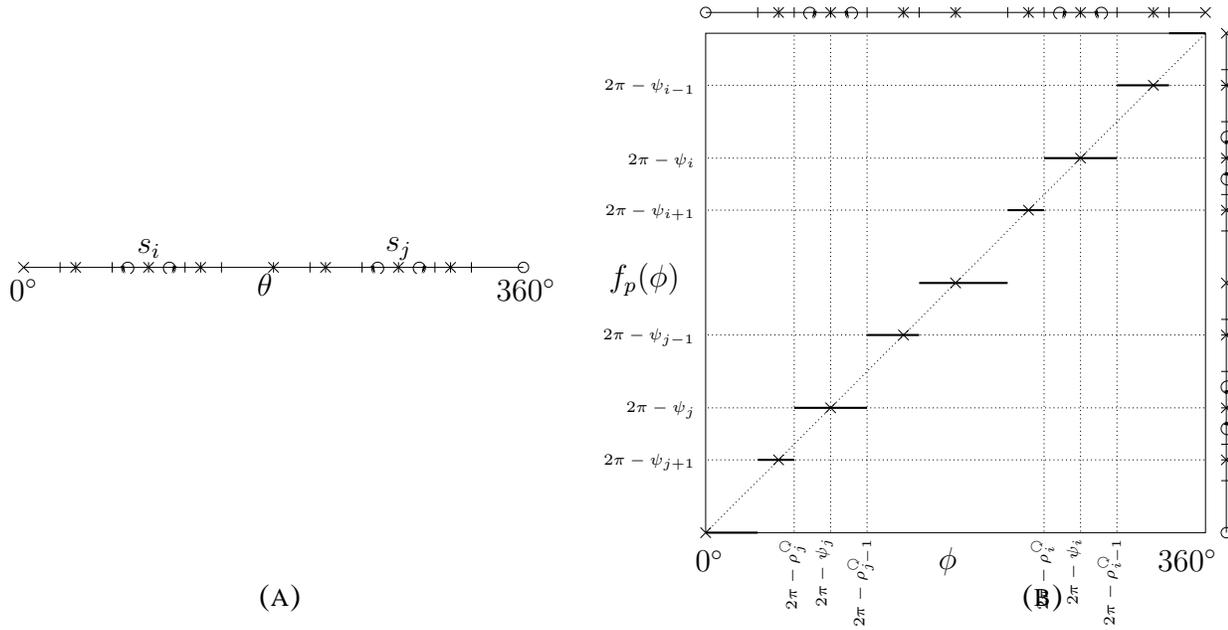
(A)   (B)

PSfrag replacements

Figure 14: THE PUSH FUNCTION AND ITS RELATIONSHIP WITH SENSOR ACTION RANGES. (A) SHOWS THE RESTING RANGES AND (B) SHOWS THE PUSH FUNCTION WITH 'FLIPPED' SENSOR RESTING RANGES.
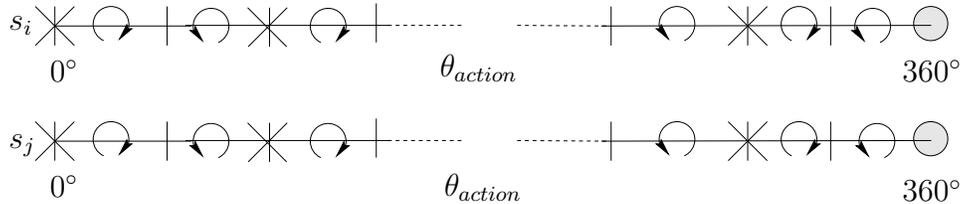
Figure 15: SENSOR ACTION RANGES ARE ALL EQUAL FOR TWO STATES $s_i \doteq s_j$. LEMMA 6.2 SHOWS THAT THIS CAN ONLY HAPPEN IF THE TWO STATES ARE EQUIVALENT WITH RESPECT TO THE PUSH FUNCTION.

to a small rotation, may be required to ensure collection of valid force state transition data. This extra step is seen in subsequent plan length calculations.

6 Maximum Plan Lengths

The length of an orienting plan is the number of push-align operations required to uniquely orient a part, and the maximum plan length is the worst case possible for a class of parts. In 6.1 we discuss the relationship between the sensor action ranges and the push function, and then characterise how the endpoints of the sensor action ranges facilitate reducing state uncertainty. In Sections 6.2 and 6.3 we present our results using only rotation direction and rotation direction plus force data respectively. In Section 6.6 we compare our results to previous work using a diameter sensor [AM99].

6.1 Critical Property of the Sensor Action Ranges

The proofs for maximum plan length are straightforward, and are in fact a direct consequence of the relationship between the sensor action ranges and the push function. In Figure 14 we see how the push function and sensor resting ranges are related. Figure 14(a) shows the resting ranges for a part with equivalent states $s_i \doteq s_j$ (refer

to Definition 4.2). The push function for the same part is shown in Figure 14(b). The sensor resting ranges are placed above and to the right of the push function after being transformed ('flipped') by the push function identity of Equation 2 (notice how $0°$ of each sensor resting range diagram is aligned with the top right corner). There is a 1:1 relationship between the steps on the f_p axis and stable orientations ψ_i, and a 1:1 relationship between step transitions on the ϕ axis and sensor resting range endpoints $\rho_i^{\circlearrowleft}$.

Next, observe in Figure 15 that if all sensor action ranges for two stable states s_i and s_j are equal then there is no action that will result in possible clockwise or counterclockwise rotation, so no action can reduce uncertainty between the two states when using only rotation direction. Lemma 6.2 proves that this only happens if the states are equivalent with respect to the push function.

In Lemma 6.1 we derive a useful intermediate condition that characterises when all the sensor action ranges are equal for two states. The parameter $z \in \mathbb{Z}$ in the following proofs is used to index or 'step along' the sensor action ranges of Figure 15.

Lemma 6.1 All sensor action ranges for two stable states s_i and s_j are equal if and only if $\psi_{j+z} - \psi_{i+z} = \rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft} = \psi_j - \psi_i$ for all $\mathbb{Z}$.
Proof Assume all the endpoints of the sensor action ranges are equal. We then have the following two relations (remembering $\mathrm{mod}\ 2\pi$ and $\mathrm{mod}\ n$ arithmetic).

$$\hat{\psi}_{i,(i+z)} = \hat{\psi}_{j,(j+z)} \qquad \text{and} \qquad \hat{\rho}_{i,(i+z)}^{\circlearrowleft} = \hat{\rho}_{j,(j+z)}^{\circlearrowleft}$$

for all $z \in \mathbb{Z}$. Using the identities (a) and (c) in Equation 3 results in

$$\psi_{i+z} - \psi_i = \psi_{j+z} - \psi_j \qquad \text{and} \qquad \rho_{i+z}^{\circlearrowleft} - \psi_i = \rho_{j+z}^{\circlearrowleft} - \psi_j$$

from which we get the relation

$$\psi_{j+z} - \psi_{i+z} = \rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft} = \psi_j - \psi_i \tag{4}$$

The other implication direction comes from the reverse mathematical steps. ∎

Now we prove a key lemma which states that all sensor action ranges for two stable states s_i and s_j are equal if and only if the states are equivalent with respect to the push function $s_i \doteq s_j$.

Lemma 6.2 $\psi_{j+z} - \psi_{i+z} = \rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft} = \psi_j - \psi_i$ for all $z \in \mathbb{Z}$ if and only if $s_i \doteq s_j$.
Proof First assume $\psi_{j+z} - \psi_{i+z} = \rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft} = \psi_j - \psi_i$, for all $z \in \mathbb{Z}$. Substituting the push function identity (Equation 2) and rearranging gives $f_p(2\pi - \psi_{i+z}) - f_p(2\pi - \psi_{j+z}) = \rho_{i+z}^{\circlearrowleft} - \rho_{j+z}^{\circlearrowleft} = \psi_j - \psi_i$, for all $z \in \mathbb{Z}$. This implies the distance between the steps $f_p(2\pi - \psi_{i+z}) - f_p(2\pi - \psi_{j+z})$ (on f_p axis) and the distance between the step transitions $\rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft}$ (on the ϕ axis) are all equal to $\psi_j - \psi_i$ for all $z \in \mathbb{Z}$. Trivially, the push function has a period of some $(\psi_j - \psi_i)/n$ where $n \in \mathbb{Z}+$, so $s_i \doteq s_j$.

Next assume $s_i \doteq s_j$. This implicitly implies that the push function is symmetric with period $(\psi_j - \psi_i)/n$, which implies that $f_p(2\pi - \psi_{i+z}) - f_p(2\pi - \psi_{j+z}) = \psi_j - \psi_i$. Substituting from the push function identity we get (†) $\psi_{j+z} - \psi_{i+z} = \psi_j - \psi_i$. Recalling the definition of push function symmetry $f_p(\phi + T) = (f_p(\phi) + T)\ \mathrm{mod}\ 2\pi$, we know that the distance between equivalent steps on the f_p axis ($\psi_j - \psi_i$) is the same as the distance between the those step's endpoints on the ϕ axis ($\rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft}$). This gives (‡) $\rho_{j+z}^{\circlearrowleft} - \rho_{i+z}^{\circlearrowleft} = \psi_j - \psi_i$, which when combined with (†) completes the proof. ∎

6.2   Using Only Rotation Direction

Lemma 6.2 implies that given any two states $s_i \neq s_j$, all of the sensor action ranges will not be equal. At least two sensor action ranges will result in an overlap range (and sensor representative action) that results in clockwise rotation for one state and counterclockwise for the other. We can thus determine uniquely the resulting state given any two states $s_i \neq s_j$, and immediately have Lemma 6.3 for sets of non-equivalent states.

Lemma 6.3 For any set of k non-equivalent states, there exists an action that reduces the number of uncertain states by at least one.
Proof Pick any two of the states $s_i \neq s_j$. Lemma 6.2 guarantees the existence of an action to determine uniquely the resulting state given any two states. If this action is applied, then in the worst case, $k-1$ states will result on one rotation direction (↻/↺) and the remaining state will result in the other rotation direction (↺/↻), thus based on rotation direction, we have at least one fewer uncertain resultant states.   ∎

We can now present Theorem 6.1 which was used to determine the plan lengths in the 'direction' column of Table 6.5.

Theorem 6.1 The maximum number of steps to orient a part with n stable edges and a push function with p periods to symmetry in the push function is n/p.
Proof When a part's push function has p periods, each state has p equivalent stable states. The number of uncertain, non-equivalent stable states after the first push align operation is thus n/p. By Lemma 6.2, the non-equivalent state uncertainty can be reduced by at least one at each step, resulting in plans no longer than n/p steps.   ∎

Notice that when using rotation direction alone, we can only orient to symmetry in the push function. Another interesting property of rotation direction is that when the push function is symmetric, the push-rotation direction function is also symmetric, which is not true of general push-sensor functions.

6.3   Using Rotation Direction and Force

When force is introduced into the mixture, the notion of (in)distinguishable sensor augmented states becomes important. The size of the largest indistinguishable set of sensor augmented states m is determined by comparing the force ranges of sensor augmented states for clockwise and counterclockwise rotation. Since clockwise and counterclockwise rotation may result in different sized indistinguishable sets sensor augmented states, m is taken to be the maximum over both directions. An algorithm to group states into indistinguish sets (and thus determine m) which is easily modifiable to sensor augmented states is presented in Section 5.5.1 in [Ake96].

Recall from Lemma 6.3 that rotation direction cannot be used to reduce the uncertainty between equivalent states, and thus the ability to reduce the uncertainty between equivalent states must be a property of the push-force function. We now present Lemma 6.4 which is similar to Lemma 6.3 for the case of *equivalent sets* of stable states. Together, these two Lemmas ensure that an action always exists to reduce state uncertainty by at least one state at each step.

Lemma 6.4 For any set of k equivalent states, there exists an action that reduces the number of uncertain states by at least one if the push-force function is asymmetric.
Proof A set of k equivalent stable states implies the push function is symmetric, so assume without loss of generality that f_p has a period of T_{f_p}. A part can violate push-sensor function symmetry in two ways (see Definition 4.4):
1. Its push-sensor function can be aperiodic—no period T such that $f_{ps}(\phi) = f_{ps}(\phi + T)$.
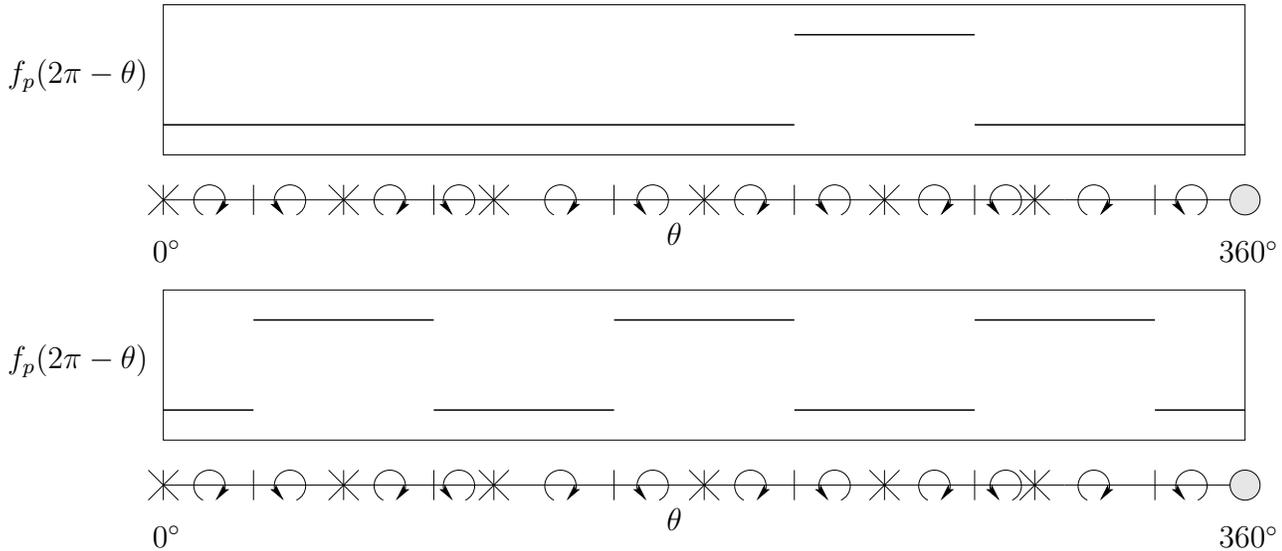
Figure 16: EXAMPLES OF ASYMMETRIC PUSH-SENSOR FUNCTIONS. (A) SHOWS AN APERIODIC PUSH-SENSOR FUNCTION AND RESTING RANGES CORRESPONDING TO A SYMMETRIC PUSH FUNCTION WITH A PERIOD OF π ($p = 2$). (B) SHOWS A PERIODIC PUSH-SENSOR FUNCTION WITH PERIOD $2\pi/3$ AND THE SAME RESTING RANGES AS IN (A).

2. Its stable states are not periodically spaced with the same period as the push-sensor function—$f_p(\phi_f + T) \neq \phi_f + T \bmod 2\pi$ for at least one stable state.

Figure 16(a) shows an example of Case (1) and Figure 16(b) shows an example of Case (2). The proof for both cases is identical.

For any two states $s_i \doteq s_j$ with stable orientations ψ_i and ψ_j, there exists an action θ_{action} that results in distinguishable sensor readings $f_{ps}(2\pi - (\psi_i + \theta_{action})) \neq f_{ps}(2\pi - (\psi_j + \theta_{action}))$. If this was not the case, then $f_{ps}(\phi)$ would by periodic with a period of T_{f_p}, and $f_{ps}(\phi)$ would be symmetric, a contradiction. Selecting this action allows us to determine uniquely the resulting state given the two states $s_i \doteq s_j$, and reduce the state uncertainty in a set of k equivalent states by at least one. ∎

We can now present Theorem 6.2 which was used to determine the plan lengths (except Case 5) in the 'direction & force' column of Table 6.5.

Theorem 6.2 The maximum number of steps to orient a part with an asymmetric push-force function, and a largest indistinguishable set of sensor augmented states of size m is $m + 1$.
Proof The worst case number of uncertain states after the first push align operation is m, where m is the size of the largest indistinguishable set of sensor augmented states. By Lemmas 6.3 and 6.4 there exists an action to reduce state uncertainty in *any* set of uncertain states by at least one at each step. Accounting for the extra step in the special degenerate case (see Section 5.2), plans can be no longer than $m + 1$ steps. ∎

If both the push and push-force functions are symmetric (Case 5 in Table 6.5), then it is impossible to distinguish between two states $s_i \doteq s_n$, since the distinguishable states are periodically spaced. We can thus only orient to symmetry in the push function, and the problem is equivalent to Case 6 with 'm' $= m/p$.

If $m = n$ the extra step is redundant and not required, however, the shortest plan for a particular part using both force and direction may still have fewer steps than when using only rotation direction. For example, the part used in Figure 2 has $m = n = 3$, but the shortest length plan when using force and rotation direction (see Figure 2(a)) has two steps (assuming the part does not contact the fence directly on a stable edge). This is better than the shortest length plan of three steps when using rotation direction alone (see Figure 2(b)).

6.4 A Polynomial Time Algorithm for Sub-optimal Plans

Since we are able to reduce the state uncertainty by at least one with each push-align operation, we are able to develop a planner with polynomial time complexity. However, the plan length is not guaranteed to be optimal. Consider a node with k uncertain states. Since an action always exists to reduce state uncertainty by at least one, in the worst case there always exists a child node with $k-1$ states.

At each level of the search we generate all the sensor representative action and select the action that results in the nodes with the fewest uncertain states. We then need only expand the child nodes of this action. As a general guideline, select the action with the minimum (over actions) maximum (over branches) number of child nodes resulting from that action. In the worst case, we need to expand nodes of size $n, n-1, \cdots, 2$ (resp. $m, m-1, \cdots, 2$). The time complexity to select the best action for a set of k states is $T_E = O(kn)$, resulting in an indistinguishable set of size $k-1$. The recurrence relation when using rotation direction is

$$T(k) = T_A(k,n) + O(kn)(T_B(k,n) + T_C(k,n) + T_D(k,n)) + T_E(k,n) + T(k-1)$$

Solving the recurrence for $T(k)$ (using Maple) and summing over $k = 2, \cdots, n$ gives $O(n^5 \log n)$. Likewise for rotation direction and force, the time complexity is $O(m^4 n \log n)$.

6.5 Completeness

The exhaustive search procedure ensures that all representative actions are applied at each level to every possible set of stable states, ensuring completeness. The termination conditions ensure that a solution is returned if one exists and it returns failure otherwise. The polynomial time algorithm is complete since at each level of the search tree, Lemmas 6.3 and 6.4 ensure the existence of an action that will reduce state uncertainty by at least one. This ensures that the algorithm will always terminate having found a solution.

6.6 Comparison to Algorithm Using Diameter Sensor

Table 6.5 presents a comparison of the two techniques in this paper with results using the diameter sensor. The diameter sensor results and descriptions of the 8 cases come from [AM99]. We can see that when using rotation direction alone, our methods outperform the diameter sensor in those cases where all the stable edges have the same diameter value (classes 7 and 8). This is due to Lemmas 6.3 and 6.4 which ensure that state uncertainty can be reduced by at least one at each step. The diameter sensor cannot achieve this in general, and in fact is equivalent to the sensorless case for classes 7 and 8. When using rotation direction and force together, our method outperforms the diameter sensor in a further three cases (classes 4, 5 and 6). In these cases, when using a diameter sensor, it is possible that available representative actions cannot distinguish between any states in a single step. Using rotation direction with the force data ensures that we can always reduce uncertainty by at least one state at each step, leading to worst case plans with fewer steps. It should also be noted that using rotation direction and force together is one step worse than the diameter sensor in the remaining three cases (classes 1, 2, and 3). The extra step arises from accommodating the special degenerate case when no rotation state transition data is collected in the first push. We also make the observation that using the rotation sensor in addition to the diameter sensor would improve the results in [AM99] for classes 4, 5, 6, 7, 8.

Akella and Mason [AM99] show that an infinite number of similarly shaped parts can have the same push function and an infinite number of similarly shaped parts can have the same push-diameter function. They use these result to characterise when multiple parts can be oriented using identical plans. Our results using only rotation direction are applicable to orienting multiple parts with the same push function, but we make no claim that multiple parts can have the same push-force function.

Table 2: MAXIMUM LENGTH OF ORIENTING PLANS. n REFERS TO THE NUMBER OF STABLE STATES, m REFERS TO LARGEST SET OF INDISTINGUISHABLE SENSOR AUGMENTED STATES, m_d REFERS TO THE LARGEST SET OF INDISTINGUISHABLE STATES (BASED ON DIAMETER), AND p IS THE NUMBER OF PERIODS IN THE PUSH FUNCTION.

	part class—(*italics* is implied but not stated in [AM99])	direction[a]	direction & force[b]	diameter[c]
1	All unique sensor values, *asymmetric or symmetric push function*	n or n/p	2	1
2	Some unique sensor values, *asymmetric or symmetric push function*	n or n/p	$m+1$	m_d
3	Multiple sensor values, none unique, aperiodic push-sensor function, *asymmetric or symmetric push function*	n or n/p	$m+1$	m_d
4	Multiple sensor values, none unique, quasi-symmetric push-sensor function, *asymmetric or symmetric push function*	n or n/p	$m+1$	$2m_d - 1$
5[d]	Multiple sensor values, symmetric push-sensor function and symmetric push function	n/p	$m/p+1$	$2m_d/p - 1$
6	Multiple sensor values, symmetric push-sensor function and asymmetric push function	n	$m+1$	$2m_d - 1$
7	Single sensor value, asymmetric push function	n	n	$2n - 1$
8[d]	Single sensor value, symmetric push function	n/p	n/p	$2n/p - 1$

[a]Ignore comments about unique/multiple/single sensor values since there are always two sensor values. "n or n/p" refers to the plan length for asymmetric and symmetric push functions respectively.
[b]Use the push-force function as the push-sensor function.
[c]Use the push-diameter function as the push-sensor function.
[d]Orientation is to symmetry in the push function $f_p(\phi)$.

7 Implementation Details

The sensor used in this paper is an ATI 15/50 6-axis force/torque sensor. Rotation direction is determined from the direction in which the contact point between the pusher and the part moves as the part rotates onto a stable edge. Section 7.1.1 shows our method for determining contact location and Section 7.1.2 outlines our method to determine rotation direction. Section 7.1.3 presents a method to deal with parts having unstable edges and Section 7.2 describes the limit surface model used to calculate ideal forces between the part and fence.

In the following sections we use the notation $\mathbf{f}_m = (f_{mx}, f_{my}, f_{mz})$ and $\mathbf{m}_m = (m_{mx}, m_{my}, m_{mz})$ to represent the measured force and moment in the force/torque sensor's reference frame. The effective coordinate system of the sensor is aligned so that the x and y axis are in the horizontal plane and the z axis points into the page. All subsequent analysis assumes that forces have only x and y components (f_{mx}, f_{my}) and the moment has only a z component m_{mz}. The flat pusher is offset in the x direction and parallel to the y axis (see Figure 17 and photo in Figure 22(a)).

7.1 Determining Rotation Direction

7.1.1 Determining Contact Location

For a hard point contact (no local torques), Bicchi, Salisbury and Brock [BSB93] give a method to calculate the contact point based on intersecting the measured applied wrench axis with the contact surface. The parametric
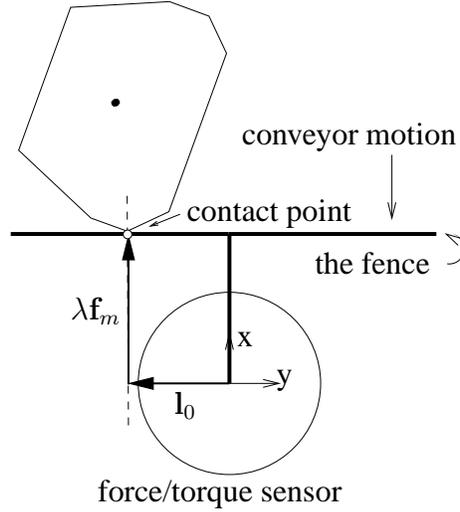
Figure 17: DETERMINING CONTACT LOCATION WITH FORCE/TORQUE SENSOR. CONTACT POINT IS AT THE INTERSECTION OF THE APPLIED WRENCH $l_0 + \lambda f_m$ AND THE FLAT FENCE. SINCE THERE IS ZERO FRICTION ASSUMED, f_m IS PERPENDICULAR TO THE FENCE.

applied wrench axis $l(\lambda)$ is determined from the measured force and moment.

$$l(\lambda) = l_0 + \lambda f_m$$

where

$$
\begin{aligned}
l_0 &= \frac{f_m \times m_m}{\| f_m \|^2} = (l_{ox}, l_{oy}) \\
&= \left(\frac{f_{my} m_{mz}}{f_{mx}^2 + f_{my}^2}, -\frac{f_{mx} m_{mz}}{f_{mx}^2 + f_{my}^2} \right)
\end{aligned}
\tag{5}
$$

The parametric line $l(\lambda)$ is then intersected with the pusher to obtain the contact point. Figure 17 shows an example assuming zero friction between the fence and part.

7.1.2   Detecting Rotation Direction

As the part rotates from vertex contact to edge contact, the contact point will move from the location of the vertex to a point on the edge of the part. This point on the edge of the part is determined by observing that after rotation onto a stable edge, the part undergoes pure translation by pushing, so the applied force must be directed through the centre of mass. With a frictionless pusher, the location of the contact point is the intersection of the fence and a perpendicular line through the centre of mass.

Figure 6 shows that as the part rotates from vertex contact to edge contact, the location of the contact point will jump from the vertex location to a point aligned with the centre of mass. Rotation direction is determined from which direction the contact point jumps. Figure 18(a)(top) shows a typical contact location profile as a part rotates onto a stable edge. At about 3.2 seconds, an edge came into contact with the pusher, and the contact point moved just over 1cm in the +y direction, indicating clockwise rotation.

Steps in the contact location profile are detected using a gradient method similar to edge detection in computer vision [GW92]. The one dimensional kernel $\hat{k} = [-1, 0, 1]$ is used to determine the gradient of a vector of data. If the contact location is stored in the vector $y[i]$, then we use the kernel $\hat{k}$ to define the gradient $g[i]$

$$g[i] = \hat{k}[0] * y[i-1] + \hat{k}[1] * y[i] + \hat{k}[2] * y[i+1]$$

(A) - STABLE EDGE

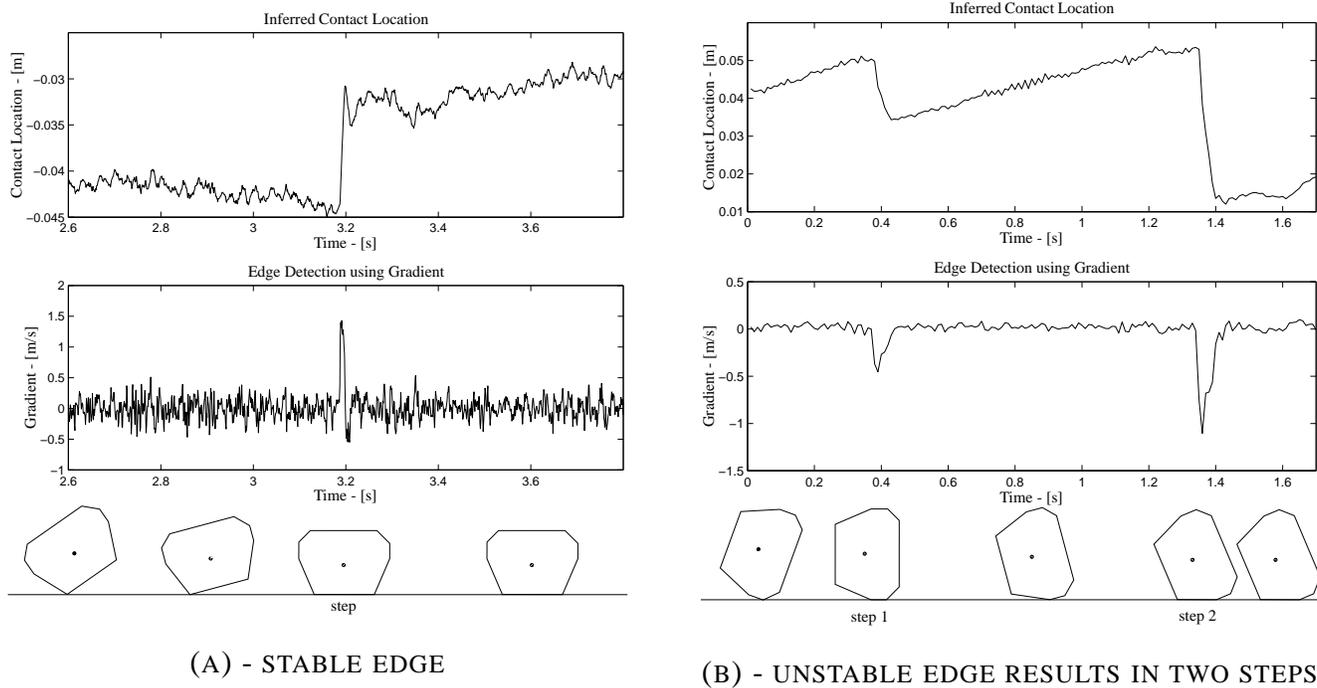(B) - UNSTABLE EDGE RESULTS IN TWO STEPS

Figure 18: LOCATION OF CONTACT POINT DURING VERTEX/EDGE TRANSITION. (TOP) SHOWS THE CHAR-ACTERISTIC STEP APPEARANCE. (MIDDLE) SHOWS THE GRADIENT USED TO DETECT THE STEPS IN THE CONTACT LOCATION PROFILE. (BOTTOM) SHOWS THE CORRESPONDING ORIENTATIONS OF THE PART.

Rotation direction is determined from the sign of the peak of the gradient. Figure 18(middle) shows the gradient of the contact location.

We emphasise that while the absolute force measurements themselves may be fairly noisy, the rotation direction is detected quite robustly (see sensitivity analysis below), because of the relatively large step change in the contact location profile. Of course, false transitions may still be detected if the pusher leaves contact with the part during the push (ie. oscillating/jittery conveyor) or if the force data is too noisy, leading to a misleading gradient, however, we have found the method to be extremely robust.

A force/torque sensor may not seem the most obvious choice for contact point detection, tactile strips or proximity sensors being more common. However, it has the overwhelming advantages of being extremely robust and well accepted in industry. As well, the force/torque sensor can be used with parts of very large mass, where the resulting forces may damage other types of contact sensor. Force/torque sensors exist that are suitable for orienting parts from a few grams to several kilograms (See the sensitivity analysis in Section 7.1.4).

7.1.3 Dealing With Unstable Edges

We now present a method to deal with parts that have one or more unstable edges. This technique requires that all stable edges are sufficiently long to be detected (see Section 7.1.4). When a part rotates onto an unstable edge, the location of the contact point moves from one vertex to another vertex, resulting in a step in the contact location. As it rotates off the unstable edge, the contact point will remain in approximately the same location. Rotation onto a stable edge results in another step in the contact location. The effect of unstable edges is to introduce multiple steps into the contact location profile, but since all stable edges are detectable, the last observed step is the stable edge. All steps will be in the same direction, so rotation direction is easily determined from any step. Figure 18(b)(top) shows the motion of the contact point for a part with an unstable edge. The gradual change (motion) in the ordinate-axis between the steps corresponds to contact slippage.

Slippage is always guaranteed assuming zero friction between the part and the fence. This slippage does not adversely affect the detection of rotation direction since the contact steps change much more rapidly. In fact, the zero friction assumption has the positive effect of increasing the rotation rate for parts, allowing for quicker orientation.

7.1.4 Sensitivity

We now address the sensitivity of this method, based on sensor accuracy and resolution. The sensitivity of contact location interpretation to force measurement noise is determined using differential methods (see [Tay82] for an excellent overview of the subject). Suppose that $x_1, ..., x_n$ are measured with uncertainties $\delta x_1, ..., \delta x_n$ and are used to compute the function $q(x_1, ..., x_n)$. If the uncertainties in the measured values are random and independent, then the uncertainty δq in q is upper bounded by

$$\delta q \leq \left| \frac{\partial q}{\partial x_1} \right| \delta x_1 + \cdots + \left| \frac{\partial q}{\partial x_n} \right| \delta x_n$$

We now use this to determine the sensitivity of the contact location to uncertainty in f_{mx}, f_{my} and m_{mz}. The location of the contact point is given by

$$(x, y) = (l_{ox} + \lambda f_{mx}, l_{oy} + \lambda f_{my})$$

Since the pusher is offset in x direction, x is constant and we have $\lambda = (x - l_{ox})/f_{mx}$. Combining and simplifying gives $y = (x f_{my} - m_{mz})/f_{mx}$, so the uncertainty in the contact location δy is

$$\begin{aligned}
\delta y &= \left| \frac{\partial y}{\partial f_{mx}} \right| \delta f_{mx} + \left| \frac{\partial y}{\partial f_{my}} \right| \delta f_{my} + \left| \frac{\partial y}{\partial m_{mz}} \right| \delta m_{mz} \\
&= \delta f_{mx} \left| \frac{x f_{my} - m_{mz}}{f_{mx}^2} \right| + \delta f_{my} \left| \frac{x}{f_{mx}} \right| + \frac{\delta m_{mz}}{|f_{mx}|}
\end{aligned} \qquad (6)$$

Notice in Equation 6 that the contact location error δy is less when (i) a larger force is applied (f_{mx} is greater) and (ii) the contact is physically closer to the origin of the force/torque sensor (x is smaller). Our apparatus could be improved by moving the fence closer to the force/torque sensor, however, some offset is inevitable since parts need to rotate unobstructed by the force/torque sensor.

For our force/torque sensor and part mass, the forces measured during orientation were on the order of 5 to 10 Newtons, corresponding to a contact location uncertainty of about ± 2mm. Our apparatus is able to consistently determine rotation direction for edges that are greater than 2cm in length. Shorter edges are also detectable, but in this case failures are more common due to sensor noise. This information allows us to present the following two rules of thumb for force/torque sensor selection:

1. Given a particular part with minimum edge length L, in order to detect the contact motion of that edge, the contact location resolution must be at least an order of magnitude finer than the edge length L.
2. In order to get contact location resolution in the millimetre range, the applied forces must be at least two orders of magnitude greater than the force sensor resolution.

7.2 Determining Ideal Forces Between Part and Fence at End of Rotation

We now present out technique to determine the ideal forces $\mathbf{f}_{ideal}$ we expect the part to exert on the fence during orientation, based on the limit surface model of quasi-static pushing [GRP91]. The limit surface model was developed in the context of a *general force* consisting of a force and an associated moment. We use the notation $\mathbf{F} = (\mathbf{f}, \mathbf{m})$ where the bold capital $\mathbf{F}$ signifies the robotics general force and $\mathbf{f}$ and $\mathbf{m}$ are the constituent force and moment.

7.2.1 Limit Surface Model of Quasi-Static Motion

Goyal, Ruina and Papadopoulos [GRP91] determined the idealised *limit surface* relationship between the applied general force $\mathbf{F} = (f_x, f_y, m)$ and the instantaneous unit velocity $\hat{\mathbf{v}} = (v_x, v_y, \omega)$ for quasi-static pushing, assuming a known pressure distribution between sliding surfaces. The limit surface is closed, convex, possibly piecewise and encloses the origin of the general force coordinate frame (see Figure 19(a)). If an applied force lies inside (resp. outside) the limit surface the object remains at rest (resp. accelerates). If the applied force lies on the limit surface, and the object is moving, the object is undergoing quasi-static motion.

Given an applied force $\mathbf{F}$ on the boundary of the limit surface, the resulting instantaneous motion is in the direction of the unit normal to the limit surface $\hat{\mathbf{v}}$. We can represent instantaneous motions of the object as a rotation about the centre of rotation (COR) $\mathbf{r}_c$, with coordinates

$$\mathbf{r}_c = (x_c, y_c) = (-v_y/\omega, v_x/\omega)$$

where $v_x{}^2 + v_y{}^2 + \omega^2 = 1$.

During part motion there are frictional forces acting between the part and the supporting surface. The frictional force $\mathbf{f}_a$ applied by the object on the supporting surface at an elemental area dA is parallel to the velocity $\mathbf{v}_a$ of that element (see Figure 19(b)). Assuming a constant pressure distribution we have

$$\mathbf{f}_a = (f_{ax}, f_{ay}) = \frac{\mu M g}{A} \left(\frac{\hat{\mathbf{k}} \times \mathbf{r}_{cor}}{\parallel \mathbf{r}_{cor} \parallel} \right) dA$$

where $\mathbf{r}_{cor} = \mathbf{r}_{cm} - \mathbf{r}_c$. The limit surface is defined in the generalised force space as the integrals of the frictional force $\mathbf{f}_a$ over the polygonal area A.

$$f_x = \int_A f_{ax} \qquad f_y = \int_A f_{ay} \qquad m = \int_A (x f_{ay} - y f_{ax}) \tag{7}$$

These relations give the force $\mathbf{F} = (f_x, f_y, m)$ that would generate the centre of rotation (x_c, y_c). However, we desire to determine the centre of rotation (from which we derive $\parallel \mathbf{f}_{ideal} \parallel$) given the part orientation and direction of the applied force $\mathbf{f}_m$—the inverse problem. We have opted to use a numerical search, and define an objective function $g(x_c, y_c)$ as the cosine of the angle between any non-zero generalised force in the direction of interest $\mathbf{F}_\perp$ ($\mathbf{f}_\perp$ perpendicular to edge and applied at vertex—see Figure 19(c)) and the numerically searched force $\mathbf{F}(x_c, y_c)$

$$g(x_c, y_c) = \frac{\mathbf{F}_\perp \cdot \mathbf{F}(x_c, y_c)}{\parallel \mathbf{F}_\perp \parallel \parallel \mathbf{F}(x_c, y_c) \parallel} \tag{8}$$

where (x_c, y_c), the location of the centre of rotation, are the search variables. The search seeks a force vector $\mathbf{F}(x_c, y_c)$ parallel to $\mathbf{F}_\perp$, which occurs at the maximum of the objective function $g(x_c, y_c) = 1$, and which a simple gradient ascent suffices to determine. The magnitude of the ideal force $\mathbf{f}_{ideal}$ is equal to the magnitude of the vector $\mathbf{f}(x_c, y_c)$ at the solution $g(x_c, y_c) = 1$.

7.2.2 Distinguishability of Force and Direction Data

Since different force measurements are possible for clockwise and counterclockwise rotation (see Figure 4), sensor measurement distinguishability is dependent on the rotation direction measurement. Figure 7 shows several ranges of force measurements corresponding to several stable states. The centre of each range $\parallel \mathbf{f}_{ideal} \parallel$ is determined using the limit surface model. The states with force ranges that overlap are considered to have
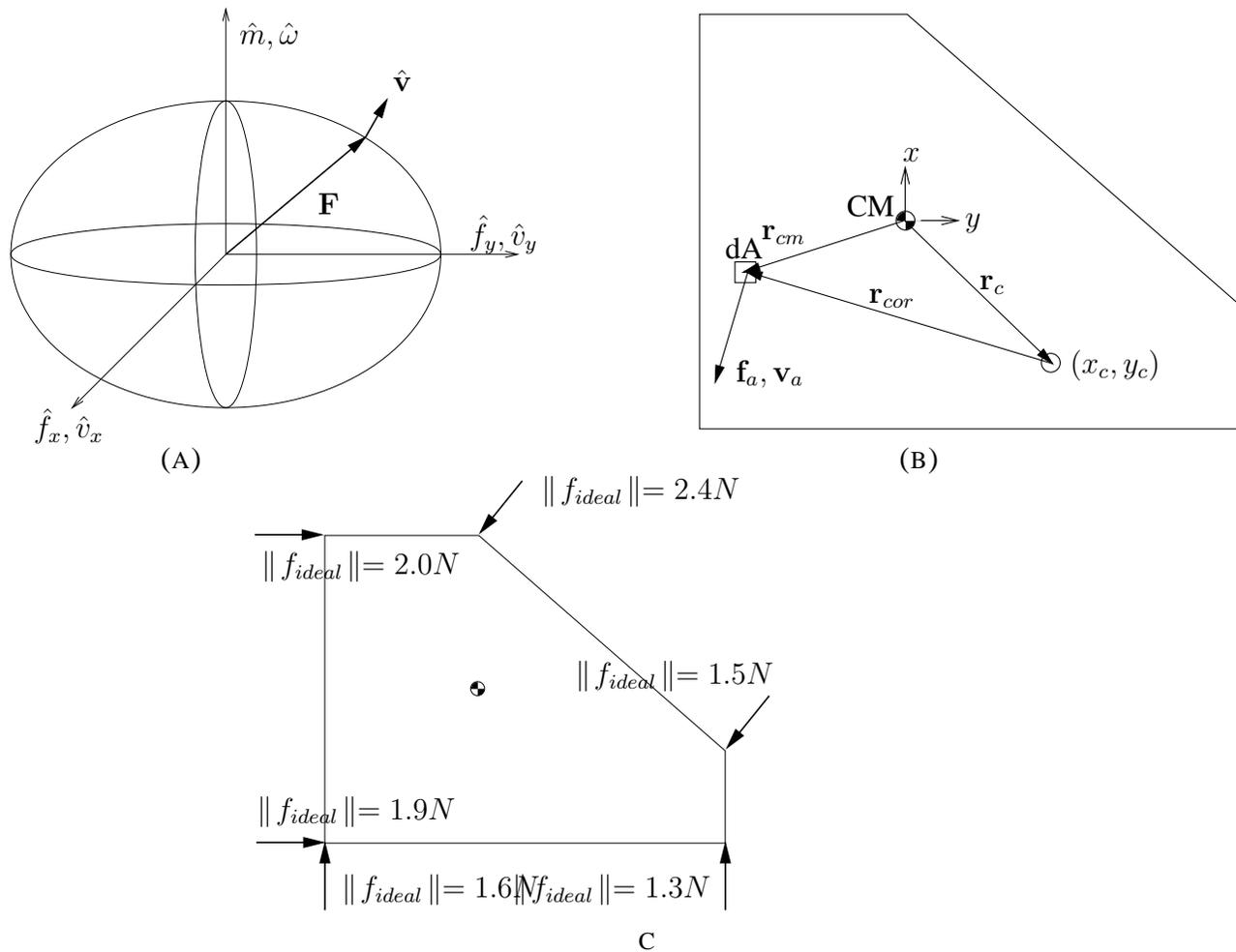
FIGURE 19: CALCULATING A TYPICAL LIMIT SURFACE. (A) SHOWS A TYPICAL LIMIT SURFACE WHERE THE INSTANTANEOUS VELOCITY $\hat{\mathbf{v}}$ ARISING FROM THE APPLIED FORCE $\mathbf{F}$ IS NORMAL TO THE LIMIT SURFACE. (B) SHOWS THE FRICTIONAL FORCE $\mathbf{f}_a$ PARALLEL TO THE VELOCITY $\mathbf{v}_a$ OF ELEMENTAL AREA dA. THE INTEGRALS OF $\mathbf{f}_a$ OVER THE SURFACE AREA A OF THE PART DEFINE THE LIMIT SURFACE. (C) SHOWS THE FORCE VECTORS $\mathbf{f}_\perp$ USED TO CALCULATE THE IDEAL FORCES $\parallel \mathbf{f}_{ideal} \parallel$. CM REFERS TO THE CENTRE OF MASS.

indistinguishable sensor readings and will all be on the same branch of the orienting plan. The range of forces represents all possible force measurements given (i) sensor noise, (ii) uncertainty in physical parameter measurement and (iii) deviations from ideal uniform pressure distribution. Bounds for (i) can be determined using Section 7.1.4. Bounds for (ii) are easily determined from measurement error when determining the constants—i.e. inclined plane angle measurement error when determining the coefficient of friction μ. Bounds for (iii) are very difficult to determine since the pressure distribution is indeterminate, and we are not aware of any studies that examine the *probability* of having any particular pressure distribution. Such bounds could be determined *empirically* based on experimentation. If the bounds are too small, leading to algorithm failures, one could simply increase the possible force ranges until the algorithm works consistently. However, if the ranges are increased too much, force information may not be useful.

8   Experimental Trials

8.1   Experimental Apparatus

An experimental testbed was built to test the algorithm. The conveyor was constructed from a pair of mylar rollers from an overhead projector. An ATI 15/50 force/torque sensor with a T-shaped fence was used to collect rotation direction and force data (see Figure 22(a)). A cable driven car on a set of tracks is used to perform the push-align operations (see Figure 22(b)). Parts are picked up and rotated using a rotating suction cup on the bottom of the car (see Figure 22(c)). The suction cup is attached to a vacuum pump, and turned on/off using a relay driven valve. The suction cup is shaped like an accordion, so when the air is removed, it actually picks the part several centimetres off the conveyor. To keep the part level and adjust the lift height, a set of four limit screws are placed around the suction cup (see Figure 22(d)).

The parts were made of aluminium, and had masses on the order of 200 grams. Our force/torque sensor has a resolution of 0.04 N/bit, so these parts only register about 20 bits of force, which results in an unusably low signal to noise ratio. In order to provide useful force readings, a 2 kg mass was added to the top of the parts. In order to pick this mass up with the suction cup, a flat surface was attached to the top of the mass. The fact that an extra mass was required to generate sufficient force should not be considered a drawback of the technique, rather it is a side effect of a force/torque sensor poorly specified for the task. Force/torque sensors with much finer resolution are available.

8.2   Measurement of Coefficient of Friction

To determine the coefficient of friction between the conveyor and the parts, the force/torque sensor was used to measure the normal force between the fence and the part during a 50 second push. The friction during the push was found to take on a large range of values, from a low of 0.3 to a high of 0.44. The average value is $\mu_{ave} = 0.37$, so the range of possible coefficients is approximately $[0.8\mu_{ave}, 1.2\mu_{ave}]$. We suggest this large range is due to an uneven distribution of dust/dirt along the conveyor, however, other factors may also be involved.

When determining the ranges of possible force measurements, to establish the distinguishability of stable states, we need to include the effect of measurement uncertainty, sensor noise and deviations from a uniform pressure distribution. The range of friction values adds $\pm20\%$ to the range, sensor noise adds about ±2-5% (0.1N over 5N) and deviations from limit surface model is placed at another $\pm20\%$. This results in ranges on the order of $\pm40\%\|\mathbf{f}_{ideal}\|$. In terms of planning, this large range of possible forces acts to increase the size of the largest indistinguishable set, often to $m = n$. This may therefore limit the usefulness of force data to
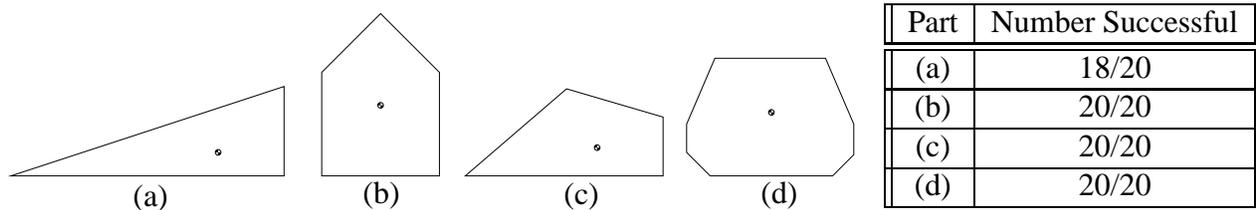
replacements

Part	Number Successful
(a)	18/20
(b)	20/20
(c)	20/20
(d)	20/20

Figure 20: CROSS-SECTIONS OF PARTS USED IN EXPERIMENTAL TRIALS USING ONLY ROTATION DIREC-TION.

PSfrag replacements

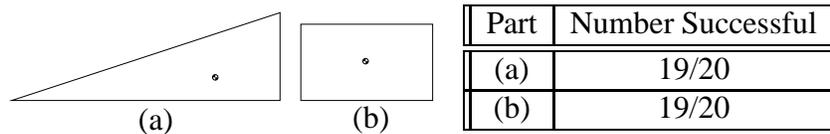| Part	Number Successful
(a)	19/20
(b)	19/20

Figure 21: CROSS-SECTIONS OF PARTS USED IN EXPERIMENTAL TRIALS USING ROTATION DIRECTION AND FORCE.

lopsided parts, i.e. parts with vertices that vary widely in their distance from the centre of mass—a hammer for instance.

8.3 Experimental Results using Only Rotation Direction

Plans generated by our algorithm were tested using parts with cross-sections shown in Figure 20. For each part, 20 trials were run. The number of successful trials is given in the second column of the table in Figure 20. Out of 80 trials, only two failures were observed. One failure occurred when the conveyor became stuck and its controller compensated by jerking it. The subsequent vibrations of the force/torque sensor lead to an invalid step detection. The second failure occurred when the suction cup failed to pick up the part. After the height of the cup was adjusted, this type of failure was not observed again.

8.4 Experimental Results using Rotation Direction and Force

For these experiments, each push-align operation is followed by a rotation direction and force reading. The rotation direction is determined from the direction of the contact location step and the force is that just *before* the step occurs. Forces during the step must not be used as they don't fit our model of vertex contact—forces during the step are usually larger than the correct forces.

The cross-sections of the two parts used are shown in Figure 21 were used for the experimental trials. When a force range of $\pm 40\% \| \mathbf{f}_{ideal} \|$ is used, the shortest length plans are no shorter than when using only rotation direction (assuming all pushes collect valid data). For this reason, the allowable deviation from uniform pressure distribution was lowered until the shortest plan length was improved. The plans generated no longer take into account possible deviations from a uniform pressure distribution, so one expects to encounter orienting failures. For part (a) this range was $\pm 30\% \| \mathbf{f}_{ideal} \|$ and for part (b) this range was $\pm 20\% \| \mathbf{f}_{ideal} \|$.

For each part, 20 trials were run using both rotation direction and force information. The results of the tests is shown in the table of Figure 21. The single failures for each part were caused by a force measurement that was slightly larger than any of the possible ranges at that stage of the plan predicted. The forces, which were approximately 0.2 Newtons too large, could have easily been accommodated with a slightly larger force range. However, increasing the force ranges would have increased the shortest plan length, rendering the force information redundant.

9 Conclusions

In this paper we have presented a novel sensor-based part orienting technique based upon the use of the commonly available force/torque sensor. Parts are oriented by a force/torque sensor equipped fence suspended above a conveyor belt and a robotic car that is able to pick up parts and rotate by a given angle. Using a sequence of push-align operations followed by force and/or rotation direction measurements, we were able to show the worst case plan lengths with the force/torque sensor are shorter than the ones in the sensorless case, and for a number of part classes, shorter than the ones for the diameter sensor used in [AM99]. A shorter plan implies a smaller number of push-align operations, hence resulting in faster execution of the plan.

The best existing sensorless techniques can orient a part in $O(n)$ steps, but have a worst case of $2n - 1$, where n is the number of stable edge of the part. The sensor-based techniques in this paper have $O(n)$ and $O(m)$ performance but result in worst case plans with fewer steps than current sensorless techniques. When using rotation direction as the sensor data, the worst case plan length is n. Using force data in conjunction with the rotation direction data reduces the worst case to $m + 1$ steps, where m is the size of the largest indistinguishable set of sensor augmented states, based on force and rotation direction.

The worst case plan length for a planner based on the diameter sensor is $2n - 1$ steps and occurs when all stable states have identical diameters. This longer than the worst case of n steps when using rotation direction. Using rotation and force, our method gives shorter plans when the push-diameter function is symmetric or quasi-symmetric. This is a direct consequence of the key property of the sensor action ranges—an action always exists that will reduce state uncertainty by at least one.

Guidelines for the selection of a force/torque sensor with appropriate resolution and range for a particular part were outlined. To detect rotation direction of an edge reliably, the contact location uncertainty must be an order of magnitude less than the corresponding edge length. To obtain contact location uncertainty in the millimetre range requires applied forces at least two orders of magnitude greater than the sensor resolution.

Since the experimental trials were successful a large majority of the time, the Coulomb model of friction used throughout this paper appears to be an adequate model for rigid, quasi-static pushing with dry sliding surfaces. However, in a manufacturing environment, the existence of dirty/oily surfaces may negatively affect the performance of these orienting techniques. Overall, rotation direction is quite reliably extracted from force measurements. Using the force magnitudes themselves, however may be practical only for heavily lopsided parts.

The force/torque sensor can provide several other pieces of useful data that may act to further reduce the number of manipulation steps. In the case of parts with unstable edges, the number of steps in the contact location profile may be useful. As well, the 'height' of the steps in the contact location offers insight into the length of the corresponding edge, or at least the relative position of the centre of mass to the edge. These data could be incorporated into the sensor action ranges and used to further distinguish final stable states during the search process.

Better planning algorithms using rotation direction and force should be developed. The additional polygonal-time algorithms presented in [Ake96] are a starting point, but algorithms with improvements directly related to the data type may be a possibility. A formalisation of the information provided by different sensor types, i.e. state and state transition sensors, should also be examined.

Acknowledgements

(a)   (b)

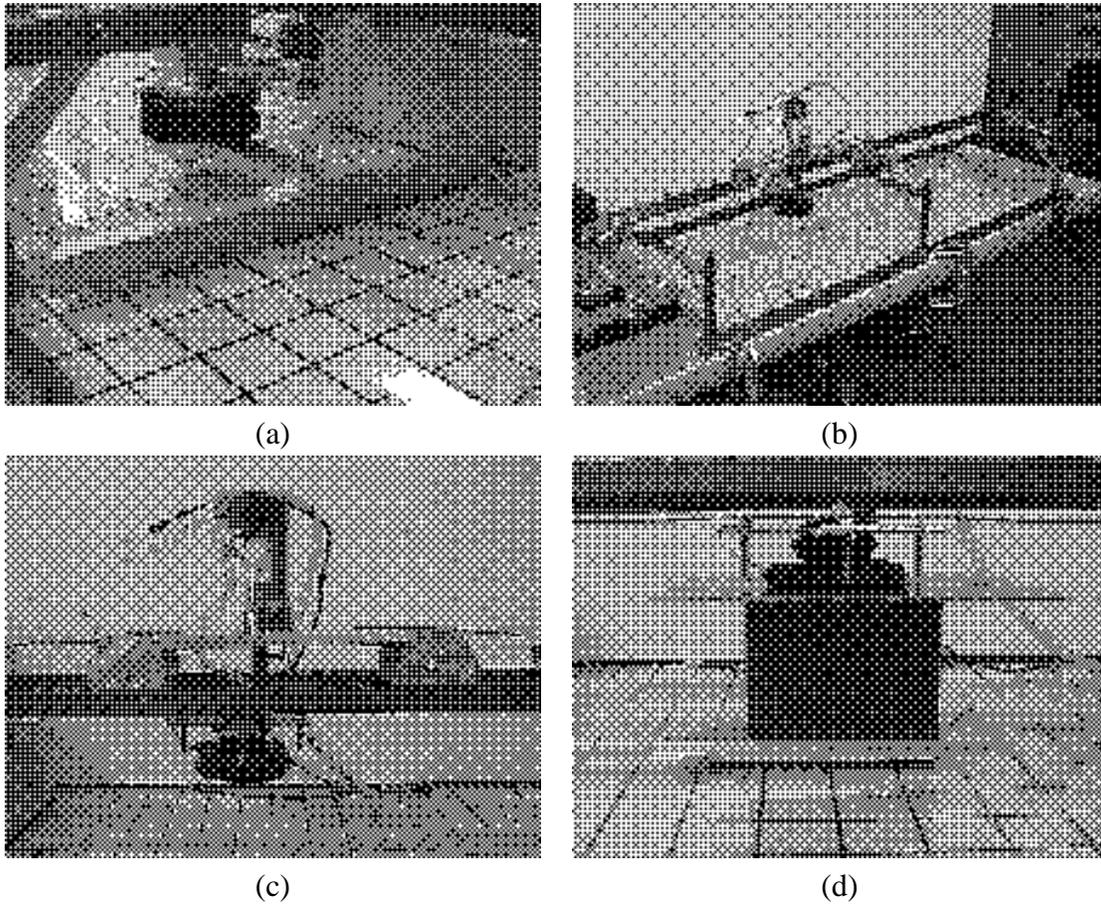(c)   (d)

Figure 22: PHOTOGRAPHS OF EXPERIMENTAL APPARATUS.

References

[AHLM97] S. Akella, W. H. Huang, K. M. Lynch, and M. T. Mason. Sensorless parts orienting with a one-joint manipulator. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2383–2390, 1997.

[Ake96] S. Akella. *Robotic Manipulation for Parts Transfer and Orientation: Mechanics Planning and Shape Uncertainty*. PhD thesis, Carnegie Mellon University, 1996.

[AM93] J. C. Alexander and J. H. Maddocks. Bounds on the friction-dominated motion of a pushed object. *International Journal of Robotics Research*, 12(3):249–262, 1993.

[AM99] S. Akella and M. Mason. Using partial sensor information to orient parts. *International Journal of Robotics Research*, 18(10):693–997, October 1999.

[BGOvdS98] R. P. Berretty, K. Goldberg, M. H. Overmars, and A. F. van der Stappen. Algorithms for fence design. In *Proc. 4th Int. Workshop on Algorithmic Foundations of Robotics (pre-prints)*, 1998.

[BPG95] M. Brokowski, M. A. Peshkin, and K. Goldberg. Curved fences for part alignment. *ASME Journal of Mechanical Design*, 117(1):27–34, 1995.

[BPM82] G. Boothroyd, C. Poli, and L. E. Murch. *Automatic Assembly*. Marcel Dekker, 1982.

[Bro88] R. C. Brost. Automatic grasp planning in the presence of uncertainty. *International Journal of Robotics Research*, 7(1):3–17, 1988.

[BSB93] A. Bicchi, J. K. Salisbury, and D. L. Brock. Contact sensing from force measurements. *International Journal of Robotics Research*, 12(3):249–262, 1993.

[CI92] Y. B. Chen and D. J. Ierardi. Oblivious plans for orienting and distinguishing polygonal parts. In *Proc. 4th Canadian Conference on Computational Geometry*, 1992.

[Gol93] K. Y. Goldberg. Orienting parts without sensors. *Algorithmica*, 10(2/3/4):201–225, 1993.

[GRP91] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction. part 1. limit surface and moment funtion. *Wear*, 143:307–330, 1991.

[GW92] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[JE96] Y. B. Jia and M. Erdmann. Pose from pushing. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 165–171, 1996.

[LM96] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability and planning. *International Journal of Robotics Research*, 15(6):533–556, 1996.

[LMT92] K. M. Lynch, H. Maekawa, and K. Tanie. Manipulation and active sensing by pushing using tactile feedback. In *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pages 416–421, 1992.

[Lyn92] K. M. Lynch. The mechanics of fine manipulation by pushing. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2269–2276, 1992.

[Mas86]   M. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3):53–71, 1986.

[MW85]   M. Mani and W. R. D. Wilson. A programmable orienting system for flat parts. In *Proceedings of North American Manufacturing Research Institute Conference XIII*, Berkeley, California, May 1985.

[PS88a]   M. A. Peshkin and A. C. Sanderson. The motion of a pushed, sliding workpiece. *IEEE Journal of Robotics and Automation*, 4(6):569–598, 1988.

[PS88b]   M. A. Peshkin and A. C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5):524–531, 1988.

[RG94]   A. S. Rao and K. Y. Goldberg. Shape from diameter: Recognizing polygonal parts with a parallel jaw gripper. *International Journal of Robotics Research*, 13(1), February 1994.

[RK91]   E. Rich and K. Knight. *Artificial Intelligence*. McGraw Hill, 1991.

[SP97]   A. Salvarinov and S. Payandeh. An application of a 1 dof fence for active part detection, manipulation and alignment on a conveyor belt. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 428–434, 1997.

[SS89]   M. Shirai and A. Saito. Parts supply in sony's general-purpose assembly system "smart". *Japan Journal of Advanced Automation Technology*, 1989.

[Tay82]   J. R. Taylor. *An Introduction to Error Analysis*. University Science Books, 1982.

[TMG87]   R. H. Taylor, M. T. Mason, and K. Y. Goldberg. *Robotics Research*, chapter Sensor-Based Manipulation Planning as a Game With Nature. MIT Press, 1987.