Transmission of microtubules through cross-channels with C_8 symmetry: A design problem based on non-linear stochastic dynamics

Cole Brooking[1], Jeff Edwards[2], Fynn Tseng[2]
[1]The Boeing Company, Seattle WA. [2]Department of Bioengineering, University of Washington, Seattle WA.

1 Abstract

'Lab on a Chip' technologies exploit the (controlled) movement of motile filaments (e.g. actin) within microfabricated devices. The major potential of this is the delivery and exchange of material between areas on the 'microfab' design. Such a design consists of a network of blocks, each of which carries out a particular function. One such block is the cross-channel, a plus-shaped junction in which microtubules effuse orthogonally. In this paper, we have investigated the effects of varying the boundaries of these cross-junctions. In particular, we optimized the transmission of microtubules across the channel such that they continued in the same direction. To do this, two different classes of boundaries were parameterized; these parameters were then varied until maximum transmission occurred. A global parameter fitting was *attempted*, but failed due to factors that will be described below.

2 Introduction

Microtubule movement is stochastic, but limited by the material stiffness properties of the tubule. The movement of the tubule as it passes through designed channels can be modeled, and likely behavior predicted using Monte-Carlo simulation. Without the ability to simulate the behavior of microtubules in a particular channel design, the design of channels for a particular purpose must be built and tested experimentally through many iterations, until the desired behavior is achieved. This process is time consuming and expensive.

In earlier work (Nitta et al, 2006), various microfabricated elements were modeled and used as the media for simulated microtubules. A good correlation was found between the modeled and experimental behaviors for straight channels, cross-junctions, concentrators, and circular tracks. In the authors' model, the cross-junction permitted $72.2 \pm 1.4\%$ of microtubules to translocate across the junction without deflection back or into one of the side channels. This particular cross-junction was perfectly rectilinear (i.e. composed of straight line segments). It became clear that the optimal design (i.e. the design with the greatest undeflected passage of microtubules) was not intuitive. We have found in our model that this is in fact *not* even a local optimum. In fact, all of our modifications presented an improvement over this channel.

The cross-junction is an integral part of sustained transport systems. The purpose is a continuous and periodic transfer of a ligand from a ligand-bound microtubule to a free microtubule. If such an interaction proceeds, the newly free filament can be cycled back into the exchanger, and so forth. The critical step in these processes is the faithful passage of the filament through, without deflecting into one of the normal channels.

Channel dimensions in a single-channel flow chamber were already shown to be important in guiding microtubules, reducing the frequency of escape, and distance between collisions (Clemmens et al, 2003). The first of these properties (guidance of microtubules) is a limiting factor in passage across a junction (such as the ones we have set out to model) because the angle of incidence, upon entering the middle of the junction, will determine the fate of the distribution of angles as it leaves. Indeed, when we modeled the path of microtubules through a straight channel (unpublished data), we saw that the frequency of final *y* positions was aberrantly high at the boundary values. By varying the curvature and concavity/convexity of the walls, we could get frequency distributions that are optimized; they would have the highest probability of final position coordinates $(x, y, \theta)_f$ that would align through the channel. An analytic approach to finding the function form of these boundaries would involve stochastic partial differential equations. A possible analytical method is hypothesized herein.

There has been little modeling of microtubule movement. The current model (Nitta et al, 2006) may be the first Monte Carlo simulation of this system. Other models have characterized the dispersion of filaments (Nitta and Hess, 2005) and the effect of stiffness on guiding as a function of the collision angle (Clemmens et al, 2003). It was found that microtubules could be bound to kinesin-coated dynodes; the diffusion away from the charged surface was modeled using Fick’s law. This model predicted, rather well, the steady-state binding of tubules to the dynode (van den Heuvel et al, 2005). The implication for our model is as follows. By controlling the deposition of filaments along channels, guidance could be controlled by an external stimulus.

Currently, the selected guidance of microtubules has evolved around rectifying channels. (Nitta et al, 2006; Hiratsuka et al, 2001). Without imposing deterministic forces on the system, such as flow (Turner et al, 1995), we desire only to minimize the stochastic effects.

3 Our Model and its Implementation

§ The Weiner process for microtubule movement

We modeled the movement of microtubules with discrete Lengevin dynamics (Nitta et al, 2006). The equations are shown below, and include persistence length, the diffusion constant, and the mean velocity as parameters. The dynamics are integrated over fixed time-steps, according to the Wiener process. Each microtubule was simulated independently, although the initial angles for the population were provided as a vector. The elements of this vector were assumed to fall along a random *flat* distribution within the prescribed range. Thus,

$$\theta_0 \subset \left[-\pi/4, +\pi/4\right] \mapsto \theta_0 = \pi/4 - (\pi/2) \cdot u(0,1)$$

The trajectories are given numerically below. It was assumed that each microtubule was a time-invariant system. The path traced by one point (the leading end) defines the path traced by the rest of the tubule. This turned out to be an important assumption, and probably an appropriate one, since the strain that is transduced along the length of the tubule would be small. The implications of this are discussed. Thus, we are allowing the stochastic processes to dominate.

$$r(t+\tau) = r(t) + \tau \cdot \left(\bar{v} + \sqrt{2D\tau} \cdot g(0,1)\right) \;\; ; \theta(t+\tau) = \theta(t) + \tau \cdot \left(\bar{\theta} + \sqrt{\bar{v}\tau/L} \cdot g(0,1)\right)$$

§ Symmetry and structure of the boundaries

The boundaries are assumed to have C_8 cyclic symmetry. This entails that the entire channel can be constructed given only one half of one channel. The coordinates of this function are then reflected across the x and y axes (i.e. to construct the vertical walls from horizontal walls, the coordinates are inverted). These walls are then concatenated to form the entire channel. We have modeled *chamfered* (linear) and *filleted* (convex circular) channels which are given by the following functional forms:

$$b_c(x) = \left\langle \begin{array}{ll} -k \cdot w & : \quad x \subset [-k,0] \\ -r/2 - x \cdot w/2 & : x \subset [-r/2, r/2] \end{array} \right. ;$$

$$b_f(x,\theta) = \left\langle \begin{array}{ll} -w \cdot [(1/2) - r + r \cdot \cos\theta] & : \theta \subset [-\pi/2, \pi/4] \\ -w \cdot [(1/2) - r + r \cdot \sin\theta] & : \theta \subset [-\pi/2, \pi/4] \end{array} \right. \cdots + b_c(x) \quad x \subset [-k,0]$$

The boundaries are parameterized by r, which appears in both functions, and controls the extent of the modification from the perfectly orthogonal cross-junction. Here, $b_c(x)$ simply defines a linear function that connects two points from orthogonal octants that are equidistant from the center of the channel. Likewise, $b_f(x)$ defines a quarter-circle with a negative radius of curvature (i.e., it bends away from the center). The boundary values for the interval $[-k,0]$ simply indicate a boundary 'primer'. This is the straight portion of the channel before the modifications. We have included it to simulate any microtubule guidance that might occur before it enters the chamfered or circular part.

§ A collision detection scheme for microtubule and wall segments

The detection of the location of collision events (along the tubule length) was not trivial. We used a simple 2D analytic method (Bourke, 1989). We have two line segments (the proximal segment of the wall, and the proximal segment of the tubule). These can be ascribed parameters for their coordinates the (x, y) space, which are then compared in the following way:

$$x_c(t) = x_0^t + \mu_a \left(x_1^t - x_o^t\right) \;\; ; \;\; y_c(t) = y_0^t + \mu_a \left(y_1^t - y_o^t\right)$$

The 'coincidence variables', μ_s and μ_b, are defined in similar ways

$$\mu_a = \left[\left(x_1^b - x_0^b\right)\left(y_0^t - y_1^w\right) - \left(y_1^b - y_0^b\right)\left(x_0^t - x_1^w\right)\right] / \left[\left(y_1^b - y_0^b\right)\left(x_1^t - x_o^t\right) - \left(x_1^b - x_0^b\right)\left(y_1^t - y_o^t\right)\right]$$
$$\mu_b = \left[\left(x_1^t - x_0^t\right)\left(y_0^t - y_1^w\right) - \left(y_1^t - y_0^t\right)\left(x_0^t - x_1^w\right)\right] / \left[\left(y_1^b - y_0^b\right)\left(x_1^t - x_o^t\right) - \left(x_1^b - x_0^b\right)\left(y_1^t - y_o^t\right)\right]$$

Collisions occur if the following Boolean function is satisfied.

$$\langle 0,1 \rangle \mapsto \mu_a \subset [0,1] \wedge \mu_b \subset [0,1]$$

The comparisons of these variables must be done pointwise because the dimensions are different:

$$\dim\left[\left(x_0^t, y_0^t\right),\left(x_1^t, y_1^t\right)\right] \neq \dim\left[\left(x_0^b, y_0^b\right),\left(x_1^b, y_1^b\right)\right]$$

Ideally, we would be able to implement the collision detection using matrix multiplication exclusively. This would have significantly accelerated the run-time.

§ A test for the successful transmission of microtubules across the junction

For each tubule, if intersections are found with any of the wall segments, the nearest intersection point is used, and all other intersections are discarded. The intersection is then inset slightly to the 'inside' side of the wall segment to avoid algorithmic complications that arise from the coincidence of the new point on the wall. Without this inset, the algorithm to maintain the microtubule 'inside' the bounds of the cross-junction would be significantly more time consuming. The dot product between the microtubule tip vector and the wall segment vector is used to set the new angle after the collision with the wall to be parallel to the wall, but not at an acute angle to the previous microtubule tip angle. If a microtubule exits the end of any of the modeled channels, it is assumed that the tubule's fate is determined, and the simulation of that microtubule terminates. If the tubule exits through the path directly opposite the inlet, the simulation is considered a success. If the tubule exits through any other path, the simulation is considered a failure. The percentage of total success is determined to be the number of successful simulations divided by the total number of simulations. In case some design acts more as a collector than a junction, trapping the microtubule, the simulation is limited to 10000 time steps (1000 seconds). If microtubules have not exited the cross junction after that time, they are considered to be failures. During these simulations, such events were very rarely encountered.

§ Actual simulated boundary parameters

For this analysis, input and output channels connected to the cross junction are defined to have a width of 4 (as in previous experiments), and the input and output channels are separated by a distance of $6 \cdot w$. To maximize the flexibility of this junction, all designs are required to be symmetrical, as above. Designs are implemented in a single octant of the design space as a series of line segments, and then mirrored about the axes of symmetry to produce a full representation of the cross junction. An additional $3 \cdot w$ length of channel is modeled at each inlet/outlet. By doing this, we ensure that the performance of the junction will be uniform if utilized from any direction. This is a limitation of our model which will be discussed below. The filleted (circular) and chamfered (linear) radii were varied according to TABLE 1 to produce the following simulated designs:

Design Type	Parameter value (w/r)	General Comments
Filleted	0.25	Each chamfered channel is composed of three linearly interpolated points.
	0.50	
	1.00	
	1.50	
	2.50	
Chamfered	0.00	Each curved function is composed of a variable number of points, but collisions are still detected using a linear interpolation.
	0.25	
	0.50	
	1.00	
	1.50	
	2.50	

### §	Behavior of microtubules during a collision

When a tubule contacts the wall, we resected the excess length and added it to the guided length. This was only done when the tubule tip was sufficiently close to the wall. Otherwise, false positive events would be detected, in which the angle is parallelized with the wall before a collision. We resect the remaining length because the mean of Weiner process of the next iteration would have an offset. In light of all of this, we may express the remaining length as a function of the unit vector in the r-direction. The angle is deflected to be parallel to the boundary; we approximate the tangent between the two nearest points.

$$r_c = r(t) - \left|\left\langle x_c(t) - x(t-\tau), y_c(t) - y(t-\tau)\right\rangle\right| \cdot \vec{r} \; ; \; \theta_c = \tan^{-1}\left[\left(y_1^b - y_0^b\right)/\left(x_1^b - x_0^b\right)\right]$$

In other words, we have computed the norm of the a vector, which represents the difference of the collision position with the previous tip position, and multiplied it by the unit vector to give us radial position at which the collision occurs.

## 4	Results

The simulations proposed by the paper are able to replicate previously performed experimental results for several geometric channel features, each serving a particular functional purpose in the network. In this report, we concentrate on a particular feature: the orthogonal cross junction. A simple 4-way junction, this feature allows tubules to enter from two directions, and leave in two directions. The prefered behavior is for mirotubules to exit through the passage directly opposite their entry passage, thus allowing for predictable behavior of mirotubules passing from either direction. Based on previous simulations, the authors found that $82.0 \pm 2.0\%$ of tubules passed directly through the junction (Clemmens et al, 2003). While these results are similar in magnitude, there is still a 10% discrepancy between other papers which should be taken into consideration when using these simulations for experiment design. However, the later result (higher frequency of passage) was observed across 89 tubules, which is probably much less than the population tested in the modeling paper (Nita et al, 2006). A χ^2 contingency test could be used to determine the significance of this discrepancy. However, for our simulations, we have not assumed any underlying distribution *de novo*. For each design case, 2500 microtubule simulations were performed. The resulting success rates are shown TABLE 2 and the trajectories of some microtubules (showing both unsuccessful and successful transmission) are shown in FIGURE 1(a-f) and FIGURE 2(a-d).

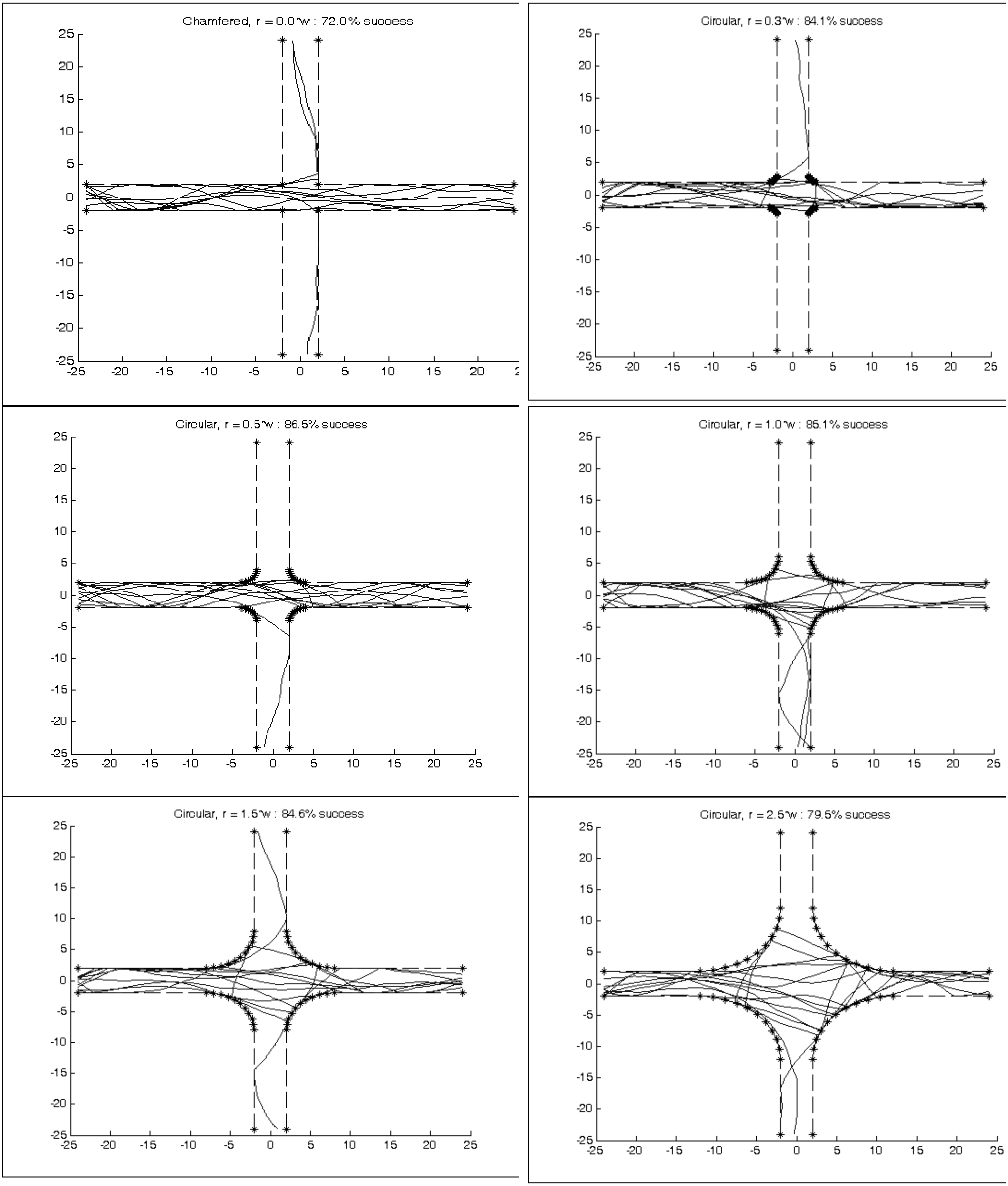
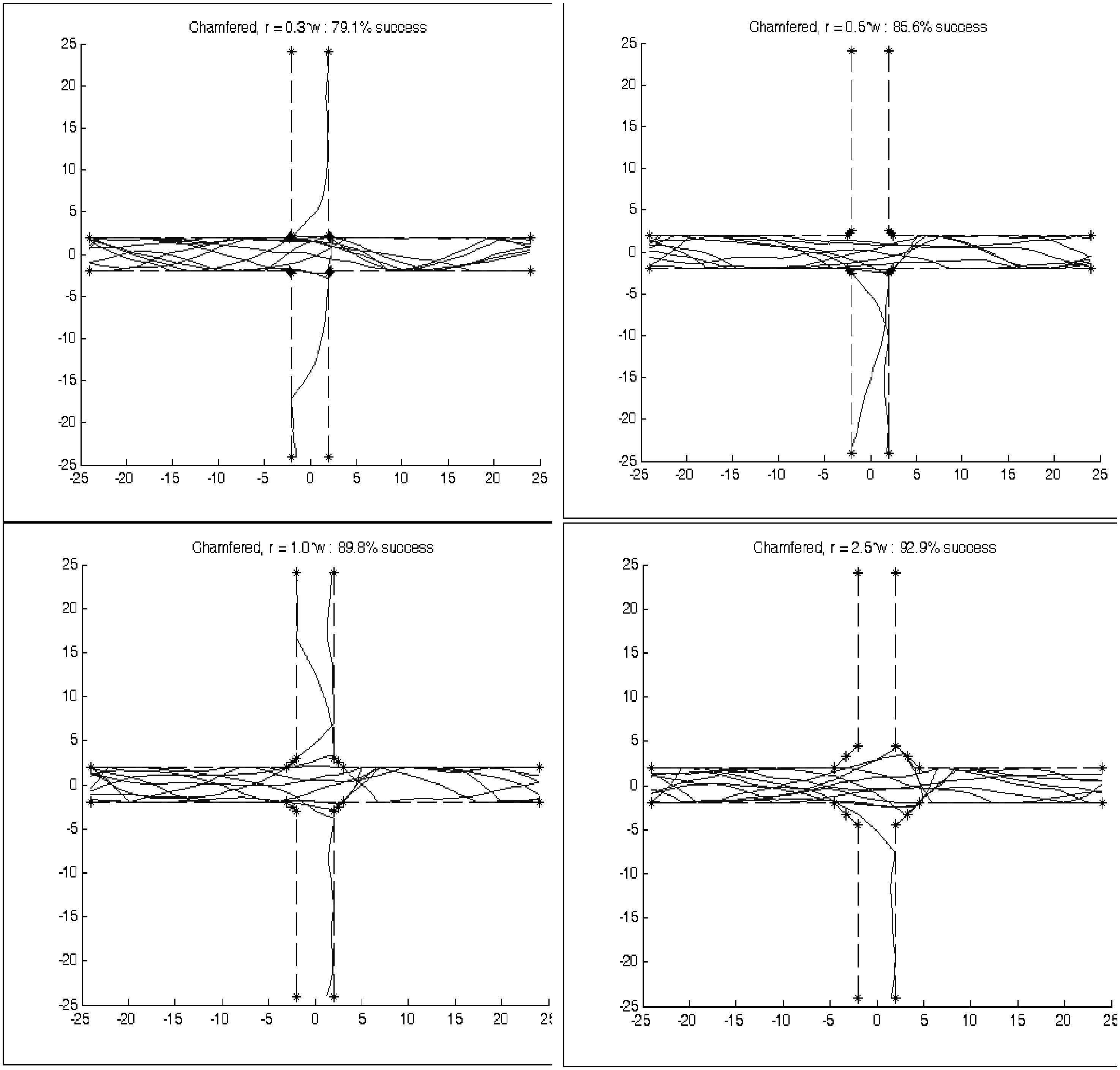


FIGURE 1 The trajectories of several microtubules through filleted (circular) cross-junction. The parameter is a non-dimensional scalar w/r . There is a high frequency of guidances in the channel primer. Microtubules enter the channel from the 'left' side, and their initial angles are uniformly distributed between two angles with zero mean, as above.


Chamfered, r = 0.3˜w : 79.1% success

Chamfered, r = 0.5˜w : 85.6% success

Chamfered, r = 1.0˜w : 89.8% success

Chamfered, r = 2.5˜w : 92.9% success

FIGURE 2 The trajectories of several microtubules through chamfered (circular) cross-junction. The rate of successful transmission is visibly higher than that in the circular channels. Each chamfered boundary is interpolated between three points, allowing collisions to be detected with arbitrarily small resolution.

These trajectories seem to indicate that the curvature of the boundary is highly influential in determining the reflected (or guided) response of the tubules. By observation, we see that the chamfered channels are more proficient at deflecting errant microtubules (those which enter the middle at extreme angles) back into the appropriate channel. Only when the inbound tubule is too far deflected does it fail to be collimated forward. The rounded channels are not as adept at this. Indeed there are instances, shown in FIGURE 1(f) where multiple reflections occur, because the wall is too convex, causing the tubule to undergo large changes in its angular position. As will be discussed below, larger radii increase the probability that this will occur. For the chamfered channels, larger radii actually decrease this probability, since there is greater surface area for capture and collimation.

Design Type	Parameter value (w/r)	Success rate over 2500 microtubules	
Filleted	0.25	84.1	
	0.50	86.5	
	1.00	85.1	
	1.50	84.6	
	2.50	79.5	
Chamfered	0.00	72.0	← Control
	0.25	79.1	
	0.50	85.6	
	1.00	89.8	
	1.50	91.6	
	2.50	92.9	← Local Maximum

5 Discussion

§ Statistical Analysis

Of the cross channel designs tested, the largest chamfered design had the best simulated performance. Performance was 92.6% successful, which was 20% higher than the simple square cross channel design simulation (72.0%). Our simulation of the simple square cross channel design also correlated well against the original paper's cited simulation statistics of 72.7+/-1.4%. However, the experimental results cited in the original paper differed from their simulated results by approximately 10%. It is therefore reasonable to expect a similar dissimilarity between our simulation and an experimental test using our proposed cross channel design.

Relevant statistical values should be determined from a χ^2 significance test. The degree of freedom could assume the role of any independent variable in our simulation (for a paired test). Based on the above data, it would be instructive to compare the control frequency with the our optimal frequency. With similar parameter values, the frequencies could be compared between the two different designs (although this measure is meaningless for the control). We also compare the most successful and least successful parameters for both types of channels. For the first test we obtained $\chi^2 = 338$ and $p \leq 0.001$. The significance of this result is due mainly to the size of our population (i.e., the denominator). For the second test (for significance between designs for each frequency value), we tested only the most disparate values (the circular and linear channels for $w/r = 2.5$). We obtained $\chi^2 = 160$ and $p \leq 0.001$. This implies that the type of design has a large impact on the success rate, even with channel primers that are of equivalent size. In the final test, we determined that each channel type could be optimized. For the filleted channels, the maximum significance was described by $\chi^2 = 17.8$ and $p \le 0.001$. For the chamfered channels, the test gave $\chi^2 = 197$ and $p \le 0.001$. Among the 'optimal' values between the channels, the filleted design bested the chamfered one with $\chi^2 = 95.0$ and $p \le 0.001$. These data indicate strongly that we have obtained a better channel geometry.

It is not clear to us whether the channels could be further improved. The frequency of transmission is not necessarily a good assay of optimality. Other system outputs include the time required for transmission and the mean variance of the tubules once they are in the interior of the channel. These could be independently optimized.

We have observed, unexpectedly, that the filleted and chamfered channels behave very differently with respect to the parameterization. When we increased the radius of the former, we found that that the frequency of transmission decreased (and significantly so, as can be seen from our statistical analysis). This is explained by the curvatures of the two surfaces. For a surface with curvature (the filleted channels), the probability of reflection into the desired exit channel will decrease, since it is harder to hit the surface at $\theta \subset [0, \pi/2]$, measured form the normal. This is because as the tubule exits the first channel, the wall curves away faster than the angle can diffuse to collide with it. However, the frequency of transmission through chamfered channels actually increased, and more significantly than that which as seen in the filleted channels. This is explained by the slower curvature (in fact, none at all) of the walls, thus increasing the probability of being guided into the desired channel.

§　Limitations of the model and its non-optimality

The major limitation of this model was the lack of a rigorous optimization process. Ideally, we would have optimized the parameters that were use to construct the boundaries. The major hindrance to this is the time required for useful statistics. In the current implementation, simulating 2500 microtubules takes approximately 1500 seconds for more complex models, such as the filleted channels. Arguably, our gratuitously high χ^2 values indicate that our population size was too large. The stochasticity of our system is intimately related to our ability to implement this sort of optimization. Most minimization algorithms evaluate the error (or any other quantity) of a function along the function gradient. For multivariate functions, such as the stochastic description of a microtubule in (x, y), the random fluctuations will inhibit the minimization process. Thus, even though a small-sample size would be give results at a faster rate, the optimization would be based on very noisy data.

Our model itself is suspect in many respects. We have used the transmission rate as the single indicator of success. However, as discussed above, this quantity gives no information about the actual efficacy of the device, because it does not measure the time distribution of filaments in the channel. The variability in time would be caused by and increased (or decreased) frequency of collisions. Our model did not investigate the frequency of one microtubule colliding with another. This frequency would have been an ideal candidate for the diagnosis of the channel in terms of its ability to catalyze the exchange of material from one tubule to another.

The assumption of linear time-invariance is based on the physical properties of the tubule. We have assumed that the tubule is composed of an infinite number of small point-masses, each of which does not exert a force on the vicinal masses. This is a good assumption when the forces caused by the kinesin are much large than the forces that could be generated by this sort of interaction. Thus, the momentum of the previous iteration (and therefore, the previous segment) has no effect on the direction of the next segment. We neglect the intertance of the fluid since the diffusion is slow.

§ The inverse problem for microtubule guidance

We wish to solve, numerically, the functional form of a C_8-symmetric cross-channel that gives the highest frequency of transmission, neglecting the time this takes. The solution should be constrained by the dimensions of the channel only (i.e., if the dimensions are too large, the random walks would disperse the tubules across the entire device, leading to a loss of collimation). This problem can be framed in the context of a stochastic differential equation in which we want to exhibit a certain behavior from a set of boundary conditions. One way to do this is to apply the discrete Martingale process at the boundaries:

$$\frac{\partial^2}{\partial x \partial y}\left(\sum_{\lambda=1}^{n} \lambda \cdot p(\chi_\lambda \mid \chi_{\lambda-1},\ldots,\chi_0)\right) \sim \cdots$$

$$\cdots \frac{\partial}{\partial y}\left(\frac{\partial}{\partial x}\right)\Bigg|_{(x,y)\subset b(x,y)}\left(\overline{p}\left\langle \sum_{i=1}^{n}\left(\int_{b_f(x-r_1x,y-r_1y)}^{b_f(x,y)} p_g(x,y,\theta)_i\, dy \mid \int_{b_f(x-rx,y-ry)}^{b_f(x,y)} p_\theta(\Delta\theta) d\Delta\theta \mid, \int_{b_f(x-rx,y-ry)}^{b_f(x,y)} p_{xy}(r_c \cos\theta, r_c \sin\theta)_i\, d\Delta\theta \mid\right), \lambda \right\rangle\right)$$

We would like to minimize, along the coordinates that form the boundary, $(x,y)\subset b(x,y)$, the conditional probability that a guiding event, p_g, occurs with the probability that the approach angle permits a collision with the boundary and the probability that the position is near enough to the boundary for a collision; these are p_θ and p_{xy}. The explicit forms of these functions have actually been determined for simple channels (Clemmens et al, 2006). The probability of a guiding event is in fact a linear function of the approach angle. The approach angle initially follows the distribution we have prescribed in the model. The authors of the abovementioned model (Clemmens et al, 2006) suggest that the guided angle (after a collision) is determined by the flexural properties of the tubule (i.e. the energy stored by bending it).

$$p_\theta(\Delta\theta) = \Delta\theta \cdot EI \cdot 2r^{-1} \Big/ \max\langle p_\theta(\Delta\theta)\rangle$$

This is also a linear function of the bending angle. Thus, given the initial condition (approach angle) and the bending angle, the second integral (for p_θ) can be numerically solved. Finally, it remains to find the probability density function p_{xy}. This is determined from the instantaneous angle and radius when the collision occurs.

§   Continuous parameterization of the boundary

Another improvement that could be made is the use of Splines, Nurbs, or canonical geometry forms to describe the cross channel geometry as a continuous surface. A Nurb would allow us to parameterize the boundary by a small set of points, indicating the amplitude of the wall as a function of x. These points would then be interpolated by a continuous curve. However, given the discrete simulation method employed here, a discretized representation of the geometry as line segments as used here probably has little effect on the solution. In fact, we found that a certain type of continuous curves (such as the convex round channel) actually decreased the frequency of collimation. It is not clear whether the device could be optimized by using a concave curvature at the boundary. Not is it clear whether the boundary need be symmetric at all. We could narrow the inlet and widen the outlet to decrease the variation of microtubules coming into the channel and collimate a wider range of microtubules coming out. However, a narrower channel entails more frequent collisions, thus increasing the dispersion. This might have an undesired effect on the final angle.

6   Concluding Comments

We consider this paper to have ultimately failed in its original goals. We initially set out to use Kalman filtering, a recursive method of minimizing error based on system and measurement noise, to reduce the uncertainty in the position of microtubules. The system noise was easily determined from the Weiner processes (it is simply a non-linear transform of the radial and angular distributions). The measurement noise was hard to define. We intended to base this noise on the quantization error involved in the A/D conversion of the signal. This type of error follows a uniform distribution with an explicit mean and variance. This error depends on the least significant bit of the converter. This would be easy to apply to an imaging device that had binary pixels (i.e. on or off). However, for grey-scale images, we could not determine the form of the error. Even if the error could be found, we supposed that its magnitude would be much less than that of the actual stochastic process, making Kalman filtering inviable. We next set out to optimize the boundaries. This was not possible since we chose a population that was too large, thereby increasing the simulation time.

The team aspect of this project was difficult due to the time constraints of the team members. With final exams, term papers, and work schedules, finding significant time to meet and work together was difficult. Email was invaluable to communicate and revise code and report iterations. Like so many research projects, the final result is not exactly the path on which we started, but we feel that we have investigated a valuable design element aspect of microtubule simulation. We have found that that the optimization problem can be solved. We have modeled some boundaries that give significantly better transmittance of filaments than the standard cross-junction. This could, in some ways, be considered a success.

7 References

1. Clemmens et al. Analysis of microtubule guidance in open microfabricated channels coated with the motor-protein kinesin. Langmuir. 19. 2006.

2. van den Heuvel et al. Electrical docking of microtubules for kinesin driven motility in nanostructures. Nano Letters. 5.2. 2005.

3. Nitta and Hess. Dispersion in active transport by kinesin powered molecular shuttles. Nano Letters. 5.7.2005.

4. Hiratsuka et al. Controlling the direction of kinesin-driven microtubulemovements along microlithographic tracks. 81. 2001.

MATLAB Scripts

§ cross_channel_demo.m

```
function cross_channel_demo()
%cross_channel_demo.m

%testing rounded cross channel


w = 4; % channel width
bounds = make_rounded_bounds( 2.5 ); sim_cc( bounds, 'Circular', 2.5 );
bounds = make_chamfered_bounds( 0 ); sim_cc( bounds, 'Chamfered', 0 );


bounds = make_rounded_bounds( 1.5 ); sim_cc( bounds, 'Circular', 1.5 );
bounds = make_rounded_bounds( 1.0 ); sim_cc( bounds, 'Circular', 1.0 );
bounds = make_rounded_bounds( 0.5 ); sim_cc( bounds, 'Circular', 0.5 );
bounds = make_rounded_bounds( 0.25 ); sim_cc( bounds, 'Circular', 0.25 );

bounds = make_chamfered_bounds( 2.5 ); sim_cc( bounds, 'Chamfered', 2.5 );
bounds = make_chamfered_bounds( 1.5 ); sim_cc( bounds, 'Chamfered', 1.5 );
bounds = make_chamfered_bounds( 1.0 ); sim_cc( bounds, 'Chamfered', 1.0 );
bounds = make_chamfered_bounds( 0.5 ); sim_cc( bounds, 'Chamfered', 0.5 );
bounds = make_chamfered_bounds( 0.25 ); sim_cc( bounds, 'Chamfered', 0.25 );
```

§ make_rounded_bounds.m

```
function boundaries = make_rounded_bounds( r )
w = 4; % channel width

boundaries = [ -6*w w/2 ]; % start of standard channel

% now make a circular path, r=2.5*w, from theta = -pi/2 to -pi/4 (sym line)


for th = -pi/2:pi/20:-pi/4
    boundaries = [boundaries; w*(-0.5-r+r*cos(th)) w*(0.5+r+r*sin(th))];
end
```

§ make_chamfered_bounds.m

```
function boundaries = make_chamfered_bounds( r );
w = 4; % channel width


boundaries = [ -6*w w/2; % start of standard channel
    -w/2-r w/2;
    -w/2-r/2 w/2+r/2];

% now make a circular path, r=2.5*w, from theta = -pi/2 to -pi/4 (sym line)
```

§ sim_cc.m

```
function sim_cc( boundaries, type, r )
n = 2500;
tic;
[success,timestep,x3,x4,y3,y4, xpaths, ypaths,thetas] = filaments(n, boundaries);
figure
hold on
for i=floor( LINSPACE(1, length(xpaths), 10) ) %1:min(10,length(xpaths))
    plot( xpaths{i}, ypaths{i} )
end
plot( [x3,x4]',[y3,y4]','--*k' );
s = sprintf( '%s, r = %.1f*w : %.1f%% success', type, r, success*100 );
title( s );
disp( s )
toc
% figure
% %timestep
% plot( [1:length(timestep)]/length(timestep),timestep/n );
% title( sprintf( '%s, r = %.1f*w : timesteps to success', type, success*100 ) );

drawnow;
```

§ filaments.m

```
function [success, timesteps, x3,x4,y3,y4, xpathsout, ypathsout, thetasout] =
filaments(Ntubules, boundaries, debug, timeout)
% 'boundaries' is a N*2 array of values
% this function returns 'success' rate

if nargin < 3
    debug = 0;
end

if nargin < 4
    timeout = 10000;
end


%%  Implementation of all Boundaries (8 walls)
% Intialize the first wall; 'boundaries' contains the channel primer
x3start = boundaries(1:end-1,1); x4start = boundaries(2:end, 1);
y3start = boundaries(1:end-1, 2); y4start = boundaries(2:end, 2);

% Concatenation for all walls (symmetries)
%    [WU, WL, EU, EL, NW, SW, NE, SE]
x3 = [x3start; x3start; -x3start; -x3start; y3start; y3start; -y3start; -y3start];
x4 = [x4start; x4start; -x4start; -x4start; y4start; y4start; -y4start; -y4start];
y3 = [y3start; -y3start; y3start; -y3start; x3start; -x3start; x3start; -x3start];
y4 = [y4start; -y4start; y4start; -y4start; x4start; -x4start; x4start; -x4start];

%% Tubule Movement (An iterator)
% Physical Parameters
Vav = 0.85;Dv = 2e-3; Lp = 111; dt = 0.1;
% Parameters of the Wiener Process
r_mean = Vav*dt; r_variance = 2*Dv*dt;
dtheta_mean = 0; dtheta_variance = Vav*dt/Lp;

% Initialize tips of tubules on a uniform distribution. Tubules are
% streamlined within theta0  = [-pi/4, pi/4] and (xt0,yt0) = (xt0, 'random
% uniform yt0) within the channel primer.
xt0 = ones(Ntubules, 1)*x3start(1);        % edge of the boundary primer
yt0 = 4*rand(Ntubules, 1) - 2;             % distrubution of yt0 between [-2,2]
thetat0 = (pi/2)*rand(Ntubules, 1) - pi/4; % distribution of thetas

T = [xt0 yt0 thetat0]; %intial values for the tubules
                      %(xt0, yt0s, thetat0) are all column vectors

status = zeros( Ntubules, 1 );
if nargout > 5
    xpaths = xt0;
    ypaths = yt0;
    thetas = thetat0;
    xpathsout = {};
    ypathsout = {};
    thetasout = {};
end
% While there are tubules, we do the following
if debug == 1
    figure
    hold on
```

```
end

count = 0;
timesteps = 0;
while length( find( status == 0 ) ) > 0

    r = r_mean + sqrt(r_variance)*randn(size(T,1),1);
    rand_dtheta = dtheta_mean + sqrt(dtheta_variance)*randn(size(T,1),1);
    T(:,3) = T(:,3) + rand_dtheta; % first step from a bounded random distrubution
(thatat0)
    dx = r.*cos(T(:,3));
    dy = r.*sin(T(:,3));

    % Collision detection
    Ttest = [T(:,1) + dx   T(:,2) + dy]; % Test values
    x1 = T(:,1); x2 = Ttest(:,1);
    y1 = T(:,2); y2 = Ttest(:,2);

    % call function to intersection tubule position change with wall
    % definition

    % Expands coordinates into square matrices for pointwise comparison
    % with boundary

    for i=1:length(x1)
        x1i = x1(i);
        x2i = x2(i);
        y1i = y1(i);
        y2i = y2(i);
        isects = [];
        for j=1:length(x3)
            x3j = x3(j);
            x4j = x4(j);
            y3j = y3(j);
            y4j = y4(j);
            den = (y4j - y3j)*(x2i - x1i) - (x4j - x3j).*(y2i - y1i);
            if den * den < eps * eps
                continue
            end
            numa = (x4j - x3j).*(y1i - y3j) - (y4j - y3j).*(x1i - x3j);
            numb = (x2i - x1i).*(y1i - y3j) - (y2i - y1i).*(x1i - x3j);
            ca = numa/den;
            cb = numb/den;
            if ca > 0 & ca < 1.0 & cb > 0 & cb < 1.0
                isects = [isects; [x3j x4j y3j y4j ca] ];
            end
        end

        if length( isects ) > 0
            [a,b] = sort( isects(:,5) );
            b = b(1);
            ca = isects(b,5);
            x3j = isects(b,1);
            x4j = isects(b,2);
            y3j = isects(b,3);
            y4j = isects(b,4);
            %intersection
            ca = ca - 1e-6; % make sure we're close to the edge, but not right on
it
```

```
            xc = x1i + ca*(x2i - x1i);
            yc = y1i + ca*(y2i - y1i);
            v = [x4j-x3j y4j-y3j];
            seg = [x2i-x1i y2i-y1i];

            if dot( v, seg ) < 0
                v = -v;
            end

            v = v./norm(v); %unit vector

            segc = [xc-x3j yc-y3j];
            segcd = norm( segc );

            remaining = r - segcd;

            Ttest(i,:) = [xc yc];
            T(i,3) = atan2( v(2), v(1) );
            if debug == 1
                plot( [x1i x2i]',[y1i y2i]', 'g' );
                plot( [x3j x4j]',[y3j y4j]','g' );
                plot( xc, yc, 'o' );
            end
        end
    end

    T(:,1:2) = Ttest;

    if nargout > 5
        xpaths = [ xpaths T(:,1) ];
        ypaths = [ ypaths T(:,2) ];
        thetas = [ thetas T(:,3) ];
    end

    fail = find( T(:,1) < x3start(1) | T(:,2) < x3start(1) | T(:,2) > -x3start(1)
);
%    status( fail ) = 1;
    T(fail,:) = [];
    if nargout > 5 & length(fail) > 0
        for f=fail'
            xpathsout{end+1} = xpaths(f,:);
            ypathsout{end+1} = ypaths(f,:);
            thetasout{end+1} = thetas(f,:);
        end
        xpaths(fail,:) = [];
        ypaths(fail,:) = [];
        thetas(fail,:) = [];
    end
    unfail_idx = find( T(:,1) > -x3start(1) );
    if nargout > 5 & length( unfail_idx ) > 0
        for f=unfail_idx'
            xpathsout{end+1} = xpaths(f,:);
            ypathsout{end+1} = ypaths(f,:);
            thetasout{end+1} = thetas(f,:);
        end
        xpaths(unfail_idx,:) = [];
        ypaths(unfail_idx,:) = [];
        thetas(unfail_idx,:) = [];
    end
```

```
%    status( unfail ) = 2;
    timesteps = [timesteps;timesteps(end)+length(unfail_idx)];
    T(unfail_idx,:) = [];

    count = count + 1;
    if count > timeout
        warning('trapped tubule')
        break %the rest are failures
    end


%     special = ones(length(x1), length(x3));
%     x1exp = diag(x1)*special;
%     x2exp = diag(x2)*special;
%     x3exp = diag(x3)*special'; x3exp = x3exp';
%     x4exp = diag(x4)*special'; x4exp = x4exp';
%     y1exp = diag(y1)*special;
%     y2exp = diag(y2)*special;
%     y3exp = diag(y3)*special'; y3exp = y3exp';
%     y4exp = diag(y4)*special'; y4exp = y4exp';
%     % Collision Test
%     den = (y4exp - y3exp).*(x2exp - x1exp) - (x4exp - x3exp).*(y2exp - y1exp);
%     numa = (x4exp - x3exp).*(y1exp - y3exp) - (y4exp - y3exp).*(x1exp - x3exp);
%     numb = (x2exp - x1exp).*(y1exp - y3exp) - (y2exp - y1exp).*(x1exp - x3exp);
%     ca = numa./den;
%     cb = numb./den;
%     % Which microtuble (i) and segment (j) collide
%     [i j] = find(ca < 1 & ca > 0 & cb < 1 & cb > 0);
%     xc = x1exp(i,j) + ca(i,j).*(x2exp(i,j) - x1exp(i,j));
%     yc = y1exp(i,j) + ca(i,j).*(y2exp(i,j) - y1exp(i,j));
%
%
%
%     dx = r*cos(T(:,3));
%     dy = r*sin(T(:,3));
%     T(:,1) = T(:,1) + dx;
%     T(:,2) = T(:,2) + dy;
end

success = timesteps(end) / Ntubules ;
```