

6.4210/6.4212

Robotic Manipulation

Midterm Exam I

October 14, 2026

Working draft — Solutions

Answer the questions in the spaces provided on the question sheets. Explain your reasoning when asked; mathematical notation or unambiguous pseudocode is acceptable. If you run out of room for an answer, make a note and continue on the empty last page, labeling the problem and part clearly. Do not tear out any pages.

You are permitted to use two sheets of paper with notes on (both sides). Do not use your phone or laptop during the exam. You will not need a calculator.

Name: _____

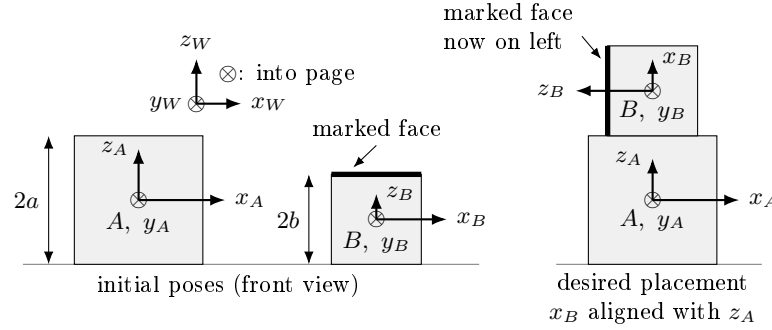
Subject (6.4210 or 6.4212): _____

MIT Kerberos (email): _____

This draft contains 6 problem(s).

1. (15 points) **Frames, grasps, and placement**

The task is for a robot to pick up cube B and place it on top of cube A . Objects A and B are cubes with side lengths $2a$ and $2b$, where $a > b > 0$. Their frames are at their centers. Initially their axes are aligned with the world axes (with $+x$ pointing to the right, $+z$ pointing upward, and $+y$ pointing into the page). Both rest on a horizontal table. The marked face of B is its $+z_B$ face.



You know the initial poses of the blocks, ${}^W X^A$ and ${}^W X^B$, and the current robot configuration q_{now} . The function

$$\text{IK}({}^W X^{G_{\text{target}}}, q_{\text{seed}}) = q_{\text{target}}$$

is available. It returns a robot configuration that puts the gripper frame G (described in part (b)) at the target pose, using q_{seed} as the solver's initial guess. Assume IK succeeds for the requested poses. Object A stays fixed.

- (a) (5 points) Given a target pose ${}^A X^{B_{\text{target}}}$ for B and a rigid grasp ${}^G X^B$, and using q_{now} as the IK seed, write the placement configuration q_{place} , at which the gripper holds B with this grasp at its target pose.

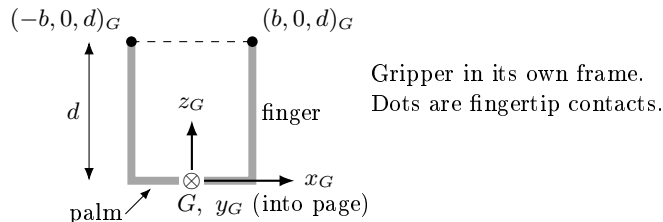
Solution:

$$q_{\text{place}} = \text{IK}({}^W X^A {}^A X^{B_{\text{target}}} ({}^G X^B)^{-1}, q_{\text{now}}).$$

Problem 1 (continued)

- (b) The parallel-jaw gripper closes along x_G ; its fingers extend along $+z_G$. The origin G is at the center of the palm, and fingertip contact points are at $(\pm b, 0, d)$ in gripper coordinates, with known $d > b$. Treat the fingers and palm as thin and assume the gripper can open far enough. Ignore the robot arm's geometry.

Give any grasp ${}^G X^B$ that can pick up B from the table and place it centered on top of A with its marked $+z_B$ face pointing left ($-x_A$) and $+x_B$ pointing upward ($+z_A$), as in the desired placement in the first figure, without changing its grasp. Specifically, for the single grasp, at both the pick and the place poses, the gripper must not touch the table or A , and only its fingers may touch B ; do not consider the motion between the two poses. Assume sufficient friction to hold B .



- i. (3 points) ${}^G R^B$ (rotation of ${}^G X^B$)

Solution:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$$

- ii. (2 points) ${}^G p^B$ (position of ${}^G X^B$)

Solution:

$$\begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}$$

TA: Contacts must be on the $\pm y_B$ faces ($x_G = \pm y_B$): a finger on $-z_B$ would touch the table at pickup, and a finger on $-x_B$ would touch A at placement. Accept any such grasp whose palm, at ${}^B p^G = -({}^G R^B)^T {}^G p^B$, is outside B and has x_B and z_B coordinates greater than $-b$ (above A at placement and the table at pickup). The other axis-aligned answer is top-down at pickup: ${}^G R^B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$, ${}^G p^B = (0, 0, d)$; its contacts are also $(0, \pm b, 0)_B$. Also accept the 180° rotation about z_G (negate rows 1-2 of ${}^G R^B$), which only swaps the two contacts, and contacts off the face centers, provided the palm rule above still holds and neither contact is on the $-x_B$ or $-z_B$ edge of its face. Grade iii on the student's own grasp.

- iii. (5 points) Give the positions where the fingertips contact cube B , in frame B .

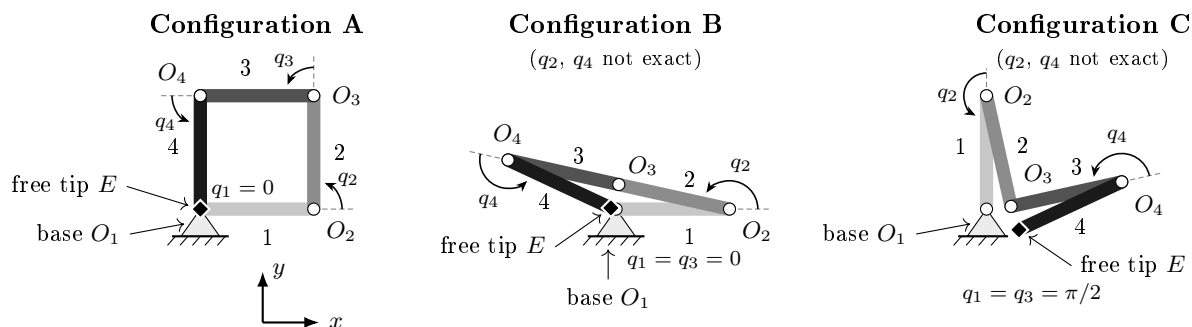
Solution: $(0, \pm b, 0)_B$, which map to $(\pm b, 0, d)_G$.

2. (15 points) **Four links good**

A planar serial arm has four unit-length links and four independently actuated revolute joints. All joint axes point out of the xy plane. Joint coordinates $q_1, \dots, q_4$ are relative angles, positive counterclockwise: q_1 is measured from $+x$, and each later angle is measured from the preceding link. The task is to control only the position $p = (x, y)$ of the free tip E , the far end of link 4, not its orientation.

In the exact configurations used for the questions, E coincides with the base, but is not attached to it. Links occupy separate parallel planes; ignore collisions.

Configurations B and C are drawn with q_2 and q_4 slightly off so that the links can be visually distinguished. Drawn exactly, B is a straight horizontal line and C an L-shape, with links overlapping. Use the exact values in the table below.



TA: Does anyone have a better idea for how to draw this, especially configuration B?

The exact planar coordinates of the joints, in link order, followed by the tip, and the joint angles are:

	O_1	O_2	O_3	O_4	E	(q_1, q_2, q_3, q_4)
A	(0, 0)	(1, 0)	(1, 1)	(0, 1)	(0, 0)	$(0, \pi/2, \pi/2, \pi/2)$
B	(0, 0)	(1, 0)	(0, 0)	(-1, 0)	(0, 0)	$(0, \pi, 0, \pi)$
C	(0, 0)	(0, 1)	(0, 0)	(1, 0)	(0, 0)	$(\pi/2, \pi, \pi/2, \pi)$

The Jacobians $(\partial p / \partial q)$ at B and C are

$$J_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad J_C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

- (a) (7 points) What is the Jacobian $(\partial p / \partial q)$ at configuration A? Hint: the Jacobian can be computed geometrically, without taking any derivatives.

Solution:

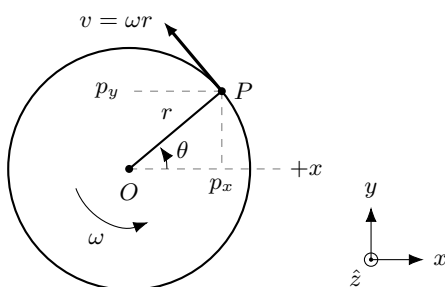
$$J_A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & -1 & -1 & 0 \end{bmatrix}.$$

Column k is the tip velocity when joint k alone turns at unit rate, $\hat{z} \times (p - O_k)$: the vector from O_k to the tip, rotated 90° counterclockwise. In A, $p - O_k = (0, 0), (-1, 0), (-1, -1), (0, -1)$, so the columns are $(0, 0), (0, -1), (1, -1), (1, 0)$. Alternatively, with $\theta_j = q_1 + \dots + q_j$, forward kinematics gives $x = \sum_{j=1}^4 \cos \theta_j$ and $y = \sum_{j=1}^4 \sin \theta_j$, so

$$\frac{\partial x}{\partial q_k} = -\sum_{j \geq k} \sin \theta_j, \quad \frac{\partial y}{\partial q_k} = \sum_{j \geq k} \cos \theta_j,$$

with $\theta = (0, \frac{\pi}{2}, \pi, \frac{3\pi}{2})$.

Additional explanation: When a rigid body rotates about an axis with angular velocity ω , a point at distance r from the axis moves with speed $v = \omega r$, perpendicular to the radius. Differentiating with respect to one angle shows where this comes from. A single joint sits at O , and a point P lies at distance r from it, at angle θ from $+x$.



Relative to O ,

$$p = (p_x, p_y) = (r \cos \theta, r \sin \theta), \quad \frac{\partial p}{\partial \theta} = r(-\sin \theta, \cos \theta) = (-p_y, p_x).$$

This vector has length r and is perpendicular to p , so turning the joint at rate $\dot{\theta} = \omega$ moves P at speed ωr , tangent to the circle.

Mapping (p_x, p_y) to $(-p_y, p_x)$ is the same as rotating p by 90° counterclockwise:

$$R\left(\frac{\pi}{2}\right)p = \begin{bmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} -p_y \\ p_x \end{bmatrix}.$$

It is also the cross product with the joint axis $\hat{z}$, which points out of the plane:

$$\hat{z} \times p = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} = \begin{bmatrix} -p_y \\ p_x \\ 0 \end{bmatrix}.$$

For the arm, turning joint k alone rotates everything beyond it rigidly about O_k , so the tip moves like P , with $p - O_k$ in place of p . That gives column k of J , $\hat{z} \times (p - O_k)$.

- (b) (3 points) For configurations B and C: state the ranks of J_B and J_C , determine if the tip can achieve an arbitrary instantaneous planar velocity, and identify whether each configuration is singular.

	Rank	Arbitrary planar velocity?	Singular?
B	1	☐ Yes ☒ No	☒ Yes ☐ No
C	2	☒ Yes ☐ No	☐ Yes ☒ No

- (c) (5 points) Kim claims: “Whenever the Jacobian has a zero column, the configuration is singular for the task.” Is Kim right for this arm?

☐ Yes ☒ No

Explain your answer.

Solution: A zero column means one joint alone produces no instantaneous tip translation. Singularity depends on all columns. J_C has zero columns 1 and 3 but full row rank: joints 2

and 4 span both task directions. (J_B also has zero columns and is singular, but because its two nonzero columns are parallel, not because of the zero columns.)

3. (25 points) **Implementing differential IK**

Consider the seven-revolute-joint Kuka arm from the coding projects. Its gripper spatial velocity, measured from the world frame W to the gripper frame G and expressed in W , is ${}^W V^G = J(q)v$, where $J(q) \in \mathbb{R}^{6 \times 7}$ and $v = \dot{q}$. The `JointVelocity` system from Problem Set 2 computes the joint-velocity command $v_{\text{cmd}} = J(q)^+ {}^W V^{G_d}$ for a desired gripper spatial velocity ${}^W V^{G_d}$.

- (a) We replace the pseudoinverse J^+ by the following Drake QP. At each control tick, J , v_{des} (${}^W V^{G_d}$), and positive joint speed limits v_{max} are known NumPy arrays of compatible shapes. The code minimizes $\|Jv - {}^W V^{G_d}\|^2$ subject to joint speed limits. To add a cost of the form $\frac{1}{2}v^T Qv + b^T v$, the arguments to `AddQuadraticCost` are Q , b , and the decision-variable vector v .

```
prog = MathematicalProgram()
v = prog.NewContinuousVariables(7, "v")
prog.AddQuadraticCost(2 * J.T @ J, -2 * J.T @ vdes, v)
prog.AddBoundingBoxConstraint(-vmax, vmax, v)
result = Solve(prog)
if not result.is_success():
    raise RuntimeError("Differential IK failed")
v_cmd = result.GetSolution(v)
```

- i. (5 points) Someone left out the constant term $\|{}^W V^{G_d}\|^2$. Is that a bug in this code?

☐ Yes ☒ No

Explain your answer.

Solution: It's not a bug. That term is constant with respect to the variables we're optimizing and so it's fine to omit it.

- ii. (5 points) How can we adjust the mathematical objective to improve behavior near singularities? Provide a formula for a new objective involving a term with multiplier λ .

Solution: $\|Jv - {}^W V^{G_d}\|^2 + \lambda\|v\|^2$, with $\lambda > 0$

- iii. (5 points) Show how to modify the code to implement this objective. Just write the line(s) to add or modify in the box below.

Solution:

```
prog.AddQuadraticCost(2 * J.T @ J + 2*lam*np.eye(7), -2 * J.T @ vdes, v)
Or keep the original cost line and add one of
prog.AddCost(lam * v.dot(v))
prog.AddQuadraticCost(2*lam*np.eye(7), np.zeros(7), v)
```

Problem 3 (continued)

- (b) Ignore the joint speed limits. In the coordinates given by the SVD of J , the objective from (a)ii splits into one independent scalar problem per singular value. For one singular value $\sigma > 0$, let $u \neq 0$ be the component of the desired gripper velocity along the corresponding task-space direction, and v the component of the joint velocity along the matching joint-space direction, so the gripper moves at σv in that task-space direction. Without regularization, the scalar problem is to minimize $(\sigma v - u)^2$; with the term from (a)ii added, its solution is

$$v_\lambda = \frac{\sigma u}{\sigma^2 + \lambda}.$$

TA: Where v_λ comes from: with regularization, the scalar problem is

$$\min_v (\sigma v - u)^2 + \lambda v^2.$$

Setting the derivative to zero,

$$\frac{d}{dv} [(\sigma v - u)^2 + \lambda v^2] = 2\sigma(\sigma v - u) + 2\lambda v = 0 \implies (\sigma^2 + \lambda)v = \sigma u \implies v_\lambda = \frac{\sigma u}{\sigma^2 + \lambda}.$$

- i. (5 points) Compare its joint speed with the unregularized solution, as σ goes to 0.

Solution: Without regularization, $v_0 = u/\sigma$. Regularization reduces the speed magnitude; for fixed $\lambda > 0$, $v_\lambda \rightarrow 0$ as $\sigma \rightarrow 0$ instead of growing without bound.

Additional explanation: $\sigma \rightarrow 0$ means the robot is approaching a singular configuration, where J is about to drop rank.

With regularization, the scalar problem is $\min_v (\sigma v - u)^2 + \lambda v^2$. Setting its derivative to zero gives $2\sigma(\sigma v - u) + 2\lambda v = 0$, so $(\sigma^2 + \lambda)v = \sigma u$ and

$$v_\lambda = \frac{\sigma u}{\sigma^2 + \lambda} = \frac{u}{\sigma + \lambda/\sigma}.$$

Without regularization, the problem is $\min_v (\sigma v - u)^2$, and $2\sigma(\sigma v - u) = 0$ gives $v_0 = u/\sigma$. For a fixed $\sigma > 0$, the regularized joint speed is slower. Since $\lambda > 0$ and $\sigma > 0$, $\lambda/\sigma > 0$, so $\sigma + \lambda/\sigma > \sigma$ and

$$|v_\lambda| = \frac{|u|}{\sigma + \lambda/\sigma} < \frac{|u|}{\sigma} = |v_0|.$$

For a fixed $\lambda > 0$, as $\sigma \rightarrow 0$,

$$|v_0| = \frac{|u|}{\sigma} \rightarrow \infty, \quad v_\lambda = \frac{\sigma u}{\sigma^2 + \lambda} \rightarrow \frac{0}{\lambda} = 0.$$

- ii. (5 points) Can regularization change the direction of the gripper's velocity, not just its speed?

☒ Yes ☐ No

Explain.

Solution: Yes. Damping cuts back motion more in directions the arm can only move slowly (near-singular, small σ) than in well-conditioned ones. When the desired velocity mixes both, the achieved velocity tilts toward the well-conditioned directions.

Additional explanation: The velocity in task space is Jv , which in the scalar problem is σv . The unregularized solution $v_0 = u/\sigma$ gives $\sigma v_0 = u$ along every direction, so the

gripper velocity equals the desired one, ratios included. The regularized one gives

$$\sigma v_\lambda = \frac{\sigma^2}{\sigma^2 + \lambda} u,$$

a factor that depends on σ . For two directions with $\sigma_1 \neq \sigma_2$, the ratio of the gripper velocity's components therefore changes from $u_1 : u_2$ to

$$\frac{\sigma_1^2}{\sigma_1^2 + \lambda} u_1 : \frac{\sigma_2^2}{\sigma_2^2 + \lambda} u_2,$$

and the near-singular, small- σ component shrinks relatively more. So the direction is kept only when every component of the desired velocity lies along a singular direction with the same σ .

TA: Accept any reason that says regularization shrinks some directions more than others, e.g. “the slow, near-singular direction gets damped more.” “No” gets no credit.

4. (20 points) **Sampling-based planning and a local extension**

Consider an RRT with straight-line extensions in configuration space.

For configurations $q_s, q_t \in \mathbb{R}^d$ ($q_{s,i}$ is coordinate i of q_s) and a positive resolution δ , consider the following path-extension procedure, which returns a success flag (**True** or **False**) and a list of configurations. If $q_s = q_t$, return (**True**, $[q_s]$). Otherwise compute the number of interpolation steps n and the interpolated configurations $q_0, \dots, q_n$, where

$$n = \max_i \left\lceil \frac{|q_{t,i} - q_{s,i}|}{\delta} \right\rceil, \quad q_j = q_s + \frac{j}{n}(q_t - q_s), \quad j = 0, \dots, n.$$

Starting at q_0 , test each successive *whole segment* $q_{j-1} \rightarrow q_j$ for collision. At the first blocked segment, return **False** and the accepted prefix ending at q_{j-1} . If all segments are free, return **True** and the whole path. Assume the start is collision-free; configurations are ordinary vectors without angle wrapping.

- (a) (5 points) Suppose an implementation checks only the interpolated configurations q_j , not the segments between them. In general, for any obstacle, is a returned path guaranteed to be collision-free?

☐ Yes ☒ No

Explain your answer.

Solution: No. A thin obstacle may lie between two free interpolated configurations. Use whole-segment collision checking, or a conservative resolution justified by geometric clearance bounds; a fixed resolution alone is not a general guarantee.

Problem 4 (continued)

- (b) (15 points) The function `extend` below grows an RRT toward q_t with the path-extension procedure above. It adds each accepted interpolated configuration q_j ($j \geq 1$) to the tree as a child of the node at q_{j-1} , and stops when the target is reached or an edge is blocked. Fill in the three blanks, (1)–(3). The following methods of `tools` are available:

`tools.nearest_node(tree, q)` Returns the tree node closest to configuration q .

`tools.edge_is_free(q_from, q_to)` Returns true exactly when the *entire closed edge* is collision-free.

`tools.add_node(tree, parent, q)` Adds a node at q as a child of `parent` and returns the new node.

A node's configuration is `node.value`. Configurations are tuples of the same dimension and may be compared with `==`. `ceil` rounds up to an integer.

`extend` returns:

- ("Reached", `node`) if the target is reached, including if it is already in the tree, with `node` the node at q_t ;
- ("Advanced", `node`) if one or more nodes were added before an edge was blocked, with `node` the last node added;
- ("Trapped", `None`) if no node could be added and the target was not reached.

def `extend(tools, tree, q_t, delta):`

```
(1)    start_node = tools.nearest_node(tree, q_t)

        q_s = start_node.value
        if q_s == q_t:
            return "Reached", start_node
        n = max(ceil(abs(q_t[i] - q_s[i]) / delta)
                for i in range(len(q_s)))
        node_prev = start_node
        for j in range(1, n + 1):

(2)    q_j = q_t if j == n else tuple(q_s[i] + (j / n) *
(q_t[i] - q_s[i]) for i in range(len(q_s)))

(3)    if not tools.edge_is_free(node_prev.value, q_j):
        if j == 1:
            return "Trapped", None
        return "Advanced", node_prev
        node_prev = tools.add_node(tree, node_prev, q_j)
    return "Reached", node_prev
```

TA: Points: 5 for each blank. Accept in blank (2) the plain formula without the $j == n$ case (landing exactly is a nicety), vector notation such as $q_s + (j/n)*(q_t - q_s)$, a list in place of a tuple, or `node_prev.value + (q_t - q_s)/n`; in blank (3) the edge from `q_s` instead of `node_prev.value`, which is equivalent here since the earlier part of that segment has passed; and in any blank a missing `tools.` prefix, or `node_prev` for `node_prev.value`, neither of which is a deduction. Do not accept in blank (2) interpolating from the moving node with fraction j/n , as in `node_prev.value + (j/n)*(q_t - node_prev.value)`: its points are not the procedure's q_j , and its steps exceed δ . From the moving node, only the constant step `node_prev.value + (q_t - q_s)/n` is right. No credit for blank (3) if the `not` is missing.

5. (25 points) **Contact wrenches**

Two fingers push on a square of side 4 cm, at $c_1 = (-1, 2)$ on the top face and $c_2 = (1, -2)$ on the bottom face. Coordinates are in cm, with the origin at the center of mass (COM). Each finger is a point contact with Coulomb friction coefficient $\mu = 1$.

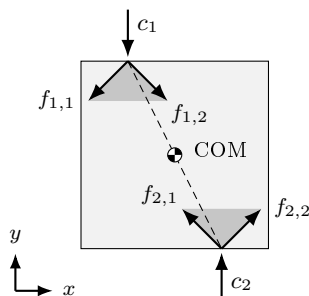
The two edges of the friction cone at contact i give two forces *applied by the finger to the object*: $f_{i,1}$, whose tangential component points left, and $f_{i,2}$, whose tangential component points right. Give each edge force an inward normal component of 1 N; since $\mu = 1$, its tangential component is also 1 N.

A force (f_x, f_y) applied at (x, y) produces the planar wrench (f_x, f_y, τ) , where

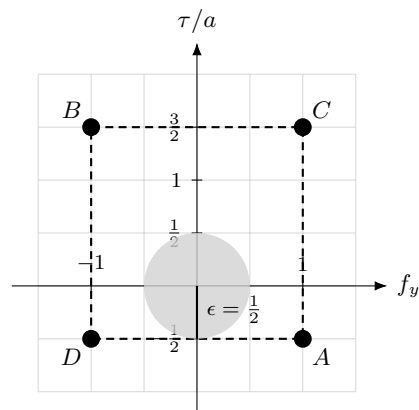
$$\tau = xf_y - yf_x$$

is the torque about the COM.

The right panel plots each edge force's wrench (f_x, f_y, τ) as the point $(f_y, \tau/a)$, labeled A–D in no particular order. Here $a = 2$ cm is half the side. Dividing by it puts both coordinates in N, which sets the scale for distances in the plot. The f_x components are omitted from the plot, but must still be included in the torque calculation.



The grasp



The edge forces' wrenches, plotted as $(f_y, \tau/a)$

- (a) (3 points) Write (f_x, f_y) for the two edge forces at the top contact, $f_{1,1}$ and $f_{1,2}$.

$$f_{1,1} = (-1, -1) \text{ N} \quad f_{1,2} = (1, -1) \text{ N}$$

Solution: Both push down; their tangential components point left and right, respectively.

- (b) (2 points) For $f_{1,1}$ at $c_1 = (-1, 2)$ cm, compute τ .

$$\tau = 3 \text{ N cm}$$

Solution: $\tau = (-1)(-1) - 2(-1) = 3 \text{ N cm}$.

Problem 5 (continued)

- (c) (10 points) The forces $f_{1,1}$ and $f_{1,2}$ act at $c_1 = (-1, 2)$ cm, and $f_{2,1}$ and $f_{2,2}$ act at $c_2 = (1, -2)$ cm. Write the point that corresponds to the wrench produced by each force. Explain briefly.

$f_{1,1}$: B $f_{1,2}$: D $f_{2,1}$: A $f_{2,2}$: C

Solution: Contact 1 pushes down, so its wrenches have $f_y = -1$ (B and D); contact 2 pushes up (A and C). With $\tau = xf_y - yf_x$: $f_{1,1} = (-1, -1)$ at $(-1, 2)$ gives $\tau = 3$ N cm, so $\tau/a = \frac{3}{2}$ (B); $f_{1,2} = (1, -1)$ gives $\tau/a = -\frac{1}{2}$ (D); $f_{2,1} = (-1, 1)$ at $(1, -2)$ gives $\tau/a = -\frac{1}{2}$ (A); $f_{2,2} = (1, 1)$ gives $\tau/a = \frac{3}{2}$ (C).

Parts (d) and (e) are for 6.4210 only.

The *convex hull* is the smallest convex region containing the four plotted points. Because the origin is strictly inside it, nonnegative combinations of the contact wrenches can produce every direction in this plotted plane.

The convex hull represents the available projected wrenches, including combinations of forces from both contacts.

- (d) (3 points) **(6.4210 only)** What is the *margin*, the radius of the largest circle centered at the origin that fits inside the convex hull?

Solution: The nearest edge is $\tau/a = -1/2$, so the margin is $1/2$ N; the sides are at distance 1 N.

- (e) (7 points) **(6.4210 only)** Why do we care about the margin, beyond whether the origin is inside the hull? What advantage does a larger margin give?

Solution: The margin is the largest disturbance magnitude that can be resisted in *every* direction in the plotted plane, using the stated force budget and scaling. A larger margin means the grasp can withstand larger disturbances even in its weakest direction; merely containing the origin does not tell us how large those disturbances can be.

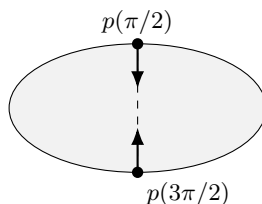
6. (10 points) **Stationary pairs and antipodal grasps (6.4212 only)**

Let $p(t)$ parameterize a smooth, simple closed planar object boundary. Let n_i be the unit *inward* normal at $p(t_i)$, and define

$$d = p(t_2) - p(t_1), \quad E(t_1, t_2) = \frac{\kappa}{2} \|d\|^2, \quad \kappa > 0.$$

Think of d as the chord between the contacts and E as the energy of a spring of stiffness κ stretched along it. A *stationary pair* is a pair (t_1, t_2) with $\nabla E = 0$. A *normal-aligned antipodal pair* (NAAP) has distinct contacts whose connecting chord points along the inward normal at each end ($n_1^T d > 0$ and $n_2^T (-d) > 0$) and is perpendicular to both tangents. Every NAAP is a stationary pair: $\partial E / \partial t_1 = -\kappa d^T p'(t_1)$ and $\partial E / \partial t_2 = \kappa d^T p'(t_2)$, and both vanish when the chord is perpendicular to both tangents. All geometry in this problem is planar.

Consider the ellipse $p(t) = (2 \cos t, \sin t)$ and $\kappa = 2$. The minor-axis pair, $(t_1, t_2) = (\pi/2, 3\pi/2)$, shown below, is a NAAP, so it is a stationary pair of E . Let t_1, t_2 vary independently. The Hessian of E at this pair, the 2×2 matrix of second derivatives of E with respect to (t_1, t_2) , has eigenvalues -4 and 12 , so it is a saddle of E .



The minor-axis pair; arrows are inward normals

Now constrain the contacts to remain diametrically opposite: $t_2 = t_1 + \pi$ (modulo 2π). The restricted energy is

$$\tilde{E}(t) = E(t, t + \pi) = 4(4 \cos^2 t + \sin^2 t).$$

- (a) (5 points) Is the minor-axis pair ($t = \pi/2$) a local minimum under this restriction? You can answer from the formula for $\tilde{E}$; no derivatives are needed.

☒ Yes ☐ No

Explain your answer.

Solution: Yes. Using $\sin^2 t = 1 - \cos^2 t$,

$$\tilde{E}(t) = 4(4 \cos^2 t + 1 - \cos^2 t) = 4(1 + 3 \cos^2 t) = 4 + 12 \cos^2 t.$$

A strict local minimum at $t = \pi/2$ means $\tilde{E}(t) > \tilde{E}(\pi/2)$ for every $t \neq \pi/2$ near $\pi/2$. Since $\cos^2 t \geq 0$, $\tilde{E}(t) \geq 4$, with equality only where $\cos t = 0$: at $t = \pi/2$, and at $t = 3\pi/2$, which is the same pair with the contacts swapped. So $\tilde{E}(\pi/2) = 4$, and $\tilde{E}(t) > 4$ for every other t near $\pi/2$.

Taking derivatives gives the same answer: $\tilde{E}'(t) = -24 \cos t \sin t$ and $\tilde{E}''(t) = -24 \cos 2t$, so $\tilde{E}'(\pi/2) = 0$ and $\tilde{E}''(\pi/2) = 24 > 0$.

- (b) (5 points) The minor-axis pair is a saddle of E . Which changes of (t_1, t_2) does the restriction rule out, and how does that explain your answer to (a)?

Solution: The restriction allows only changes with $\Delta t_1 = \Delta t_2$, where both contacts move together and the chord rotates about the center. It rules out every other change, such as $\Delta t_2 = -\Delta t_1$. Because the pair is a saddle, E decreases in some direction, but by (a) it

increases in the one allowed direction. So the restriction rules out every direction that lowers E , which is why a saddle of E is a minimum of $\tilde{E}$.

Additional explanation: Substituting $p(t) = (2 \cos t, \sin t)$ into $E = \|d\|^2$ ($\kappa = 2$) along each kind of change shows this directly.

Allowed, $t_1 = \pi/2 + s$ and $t_2 = 3\pi/2 + s$: this is $\tilde{E}(\pi/2 + s) = 4 + 12 \sin^2 s$, which is above 4 unless s is a multiple of π ($s = \pi$ gives the same pair with the contacts swapped). The chord rotates about the center and lengthens.

Ruled out, $t_1 = \pi/2 + s$ and $t_2 = 3\pi/2 - s$:

$$p(t_1) = (-2 \sin s, \cos s), \quad p(t_2) = (-2 \sin s, -\cos s), \quad d = (0, -2 \cos s),$$

so $E = 4 \cos^2 s$, which is below 4 unless s is a multiple of π . Both contacts slide sideways to $x = -2 \sin s$, so the chord stays vertical and shortens. This change is what makes the pair a saddle of E , and without it the pair is a minimum of $\tilde{E}$.

TA: A complete answer names the excluded perturbation (t_1 and t_2 changing in opposite directions, i.e. along $(1, -1)$), or says that the negative-curvature direction is excluded. “The contacts are constrained” alone is incomplete.

6.4210/2 Midterm Exam I, October 14, 2026

Additional workspace

Label each answer with its problem and part.