

Deck 05 · Optimization II: constraints, multipliers, and the KOMO formulation

4.1 Understanding constrained solutions

4.2 Solving inverse kinematics and whole motions

Özgür S. Ögüz · Bilkent University · Fall 2026

Improve the task while respecting joint limits, clearance, and exact goals.

Clipping the elbow leaves the shoulder unfinished

Same arm and target as Deck 04:

$$f(\mathbf{q}) = \frac{1}{2} \|\varphi(\mathbf{q}) - \mathbf{y}^*\|^2, \quad \mathbf{y}^* = (0.5, 0.2) \text{ m}$$

result	elbow	tip error
unconstrained solution	$\pm 80.4^\circ$	0
clip each update to the $\pm 60^\circ$ range	60°	14.6 cm

The elbow cannot bend further, but the shoulder can still reduce the error.

We need the best allowed configuration.

Separate the cost from the requirements

$$\min_{\mathbf{z} \in \mathbb{R}^d} f(\mathbf{z}) \quad \text{subject to} \quad g_i(\mathbf{z}) \leq 0, \quad h_j(\mathbf{z}) = 0$$

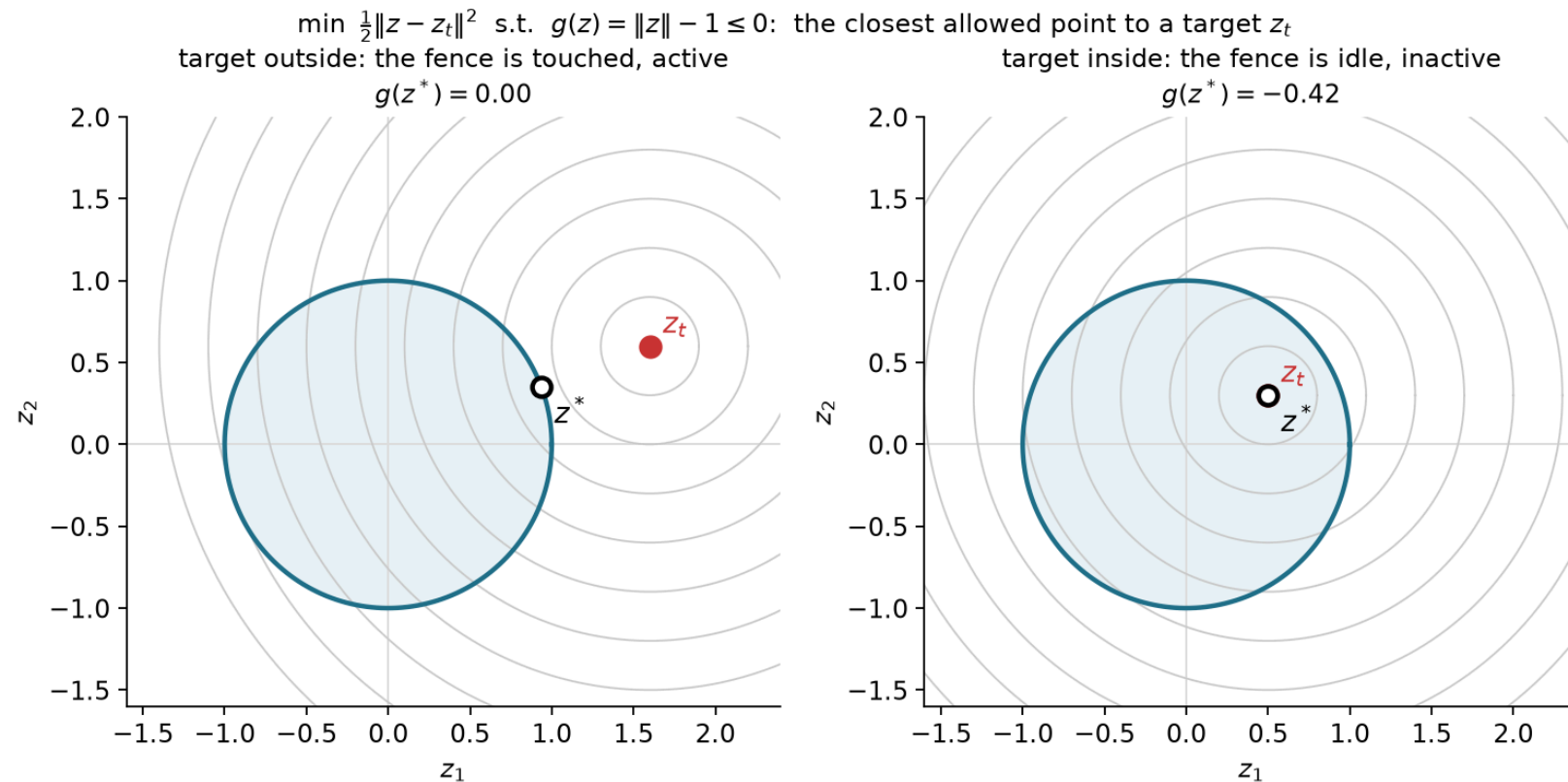
robotics requirement	constraint
joint within its range	$q_j - q_j^{\max} \leq 0, \quad q_j^{\min} - q_j \leq 0$
arm point clear of a sphere	$r_o - \mathbf{p}(\mathbf{q}) - \mathbf{c} \leq 0$
tip exactly at the goal	$\varphi(\mathbf{q}) - \mathbf{y}^* = \mathbf{0}$

$\mathbf{z}$: variables we choose; f : cost we reduce; g_i : inequality; h_j : scalar equality component.

$\mathbf{p}$: arm point; $\mathbf{c}$: sphere center; r_o : required clearance radius.

Feasible means every requirement holds. Positive g_i means a violation.

A constraint can be active or inactive



Choose the closest allowed point to the target: $\min_z \frac{1}{2} \|z - z_t\|^2$ with $g(z) = \|z\| - 1 \leq 0$.

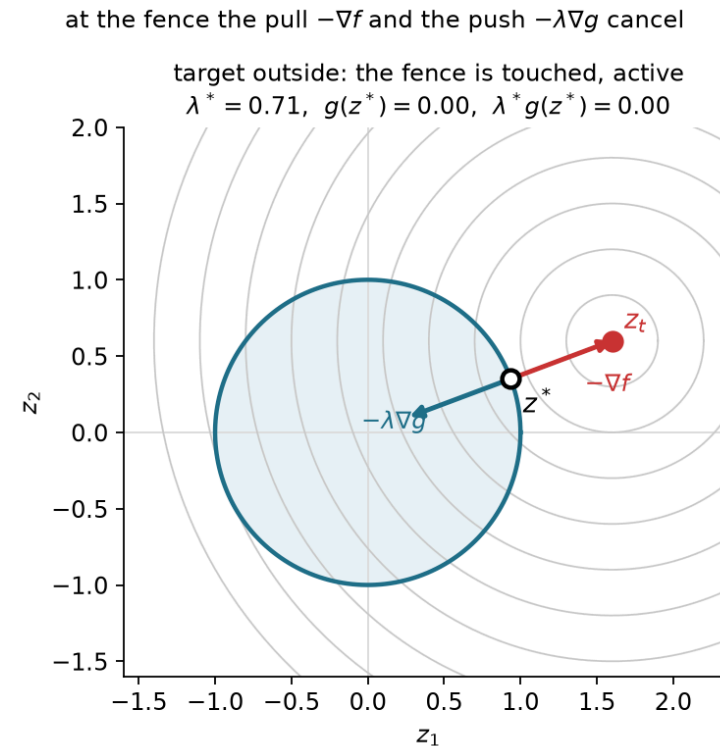
Active: on the boundary. **Inactive:** strictly inside; no restriction on this answer.

At a fence, the gradients balance

The cost favors a move through the fence. The best allowed point has no remaining downhill motion along it.

$$\nabla f(\mathbf{z}^*) + \lambda \nabla g(\mathbf{z}^*) = \mathbf{0}, \quad \lambda \geq 0$$

- $\mathbf{z}^*$: the candidate solution
- ∇g : toward the forbidden side
- λ : the **Lagrange multiplier**



In the force picture, the fence pushes inward and never pulls outward. An inactive fence has $\lambda = 0$.

Compute one multiplier from the balance

$$f(\mathbf{z}) = \frac{1}{2} \|\mathbf{z} - \mathbf{z}_t\|^2, \quad g(\mathbf{z}) = \|\mathbf{z}\| - 1 \leq 0$$

Target and closest allowed point:

$$\mathbf{z}_t = (2, 0), \quad \mathbf{z}^* = (1, 0)$$

$$\underbrace{\begin{bmatrix} -1 \\ 0 \end{bmatrix}}_{\nabla f(\mathbf{z}^*)} + \lambda \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\nabla g(\mathbf{z}^*)} = \mathbf{0} \quad \implies \quad \lambda = 1$$

Inside target: no restriction, $\lambda = 0$.

Here the multiplier equals the distance beyond the fence because of this cost and the unit constraint gradient.

Predict the effect of crossing the fence

The demo opens with the target inside.

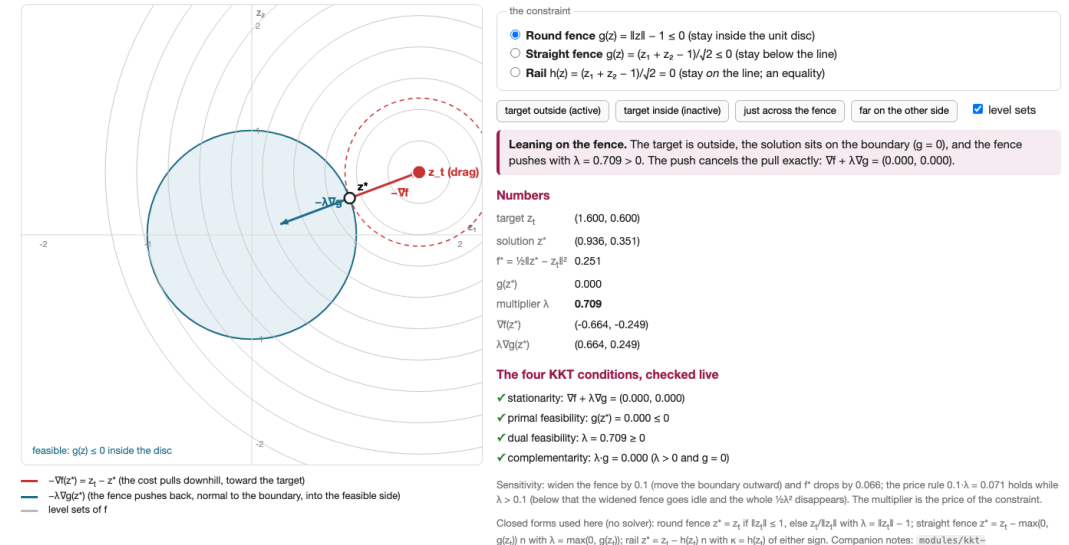
Predict: move the target to $(1.02, 0)$, just beyond the unit disc. What multiplier will balance the gradients?

Then compare a target farther outside.

Watch the allowed solution and the two balancing arrows.

KKT explorer: the fence, the force, and the multiplier

Minimise $f(z) = \frac{1}{2}\|z - z_t\|^2$ (be as close as possible to the target) while staying on the allowed side of a fence. Drag the red target. Watch the solution z^* , the downhill pull of the cost $-\nabla f$ (red), and the push of the fence $-\lambda \nabla g$ (blue) cancel each other. The multiplier λ switches on exactly when the target crosses the fence.



An exact requirement acts like a rail

- An exact goal must hold from either side: a **rail**
- A rail can oppose a pull in either direction: its multiplier κ can have either sign
- An equality must hold even when its multiplier is zero

$$h(\mathbf{z}) = 0, \quad \nabla f(\mathbf{z}^*) + \kappa \nabla h(\mathbf{z}^*) = \mathbf{0}$$

For a robot's exact position goal:

$$\mathbf{h}(\mathbf{q}) = \varphi(\mathbf{q}) - \mathbf{y}^* = \mathbf{0}$$

Each scalar equality component has its own multiplier.

Four checks on a constrained answer

$$\nabla f + \sum_i \lambda_i \nabla g_i + \sum_j \kappa_j \nabla h_j = \mathbf{0} \quad \text{gradients balance}$$

$$g_i \leq 0, \quad h_j = 0 \quad \text{all requirements hold}$$

$$\lambda_i \geq 0 \quad \text{fences push inward}$$

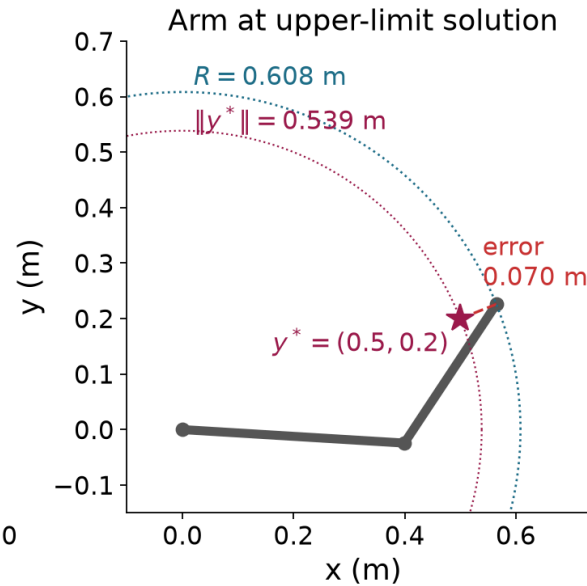
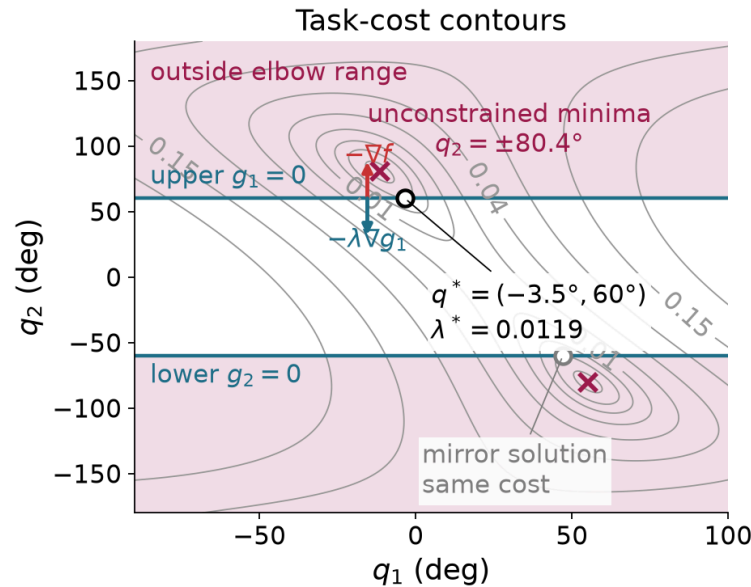
$$\lambda_i g_i = 0 \quad \text{inactive fences have no push}$$

At the candidate solution; equality multipliers may have either sign.

These are the **Karush–Kuhn–Tucker (KKT) conditions**.

Necessary at a local minimum under regularity assumptions. Passing these checks does not prove a minimum.

The constrained arm finishes the shoulder motion



result	tip error
clipped updates	14.6 cm
constrained solution	7.0 cm

Positive-elbow branch:

$$\mathbf{q}^* = (-3.5^\circ, 60^\circ)$$

The shoulder has finished improving the cost along the elbow limit.

Upper limit: active, $\lambda_1 = 0.0119 \text{ m}^2/\text{rad}$. Lower limit: inactive, $\lambda_2 = 0$.

A multiplier predicts the value of relaxing a limit

Relax one fence to $g(\mathbf{z}) \leq b$; let $f^*(b)$ be the optimized cost.

$$\frac{df^*}{db} = -\lambda$$
$$\Delta f^* \approx -\lambda \Delta b$$

Locally, the multiplier is cost saved per unit of relaxation.

A small-change prediction; the solution branch and active constraints must remain stable.

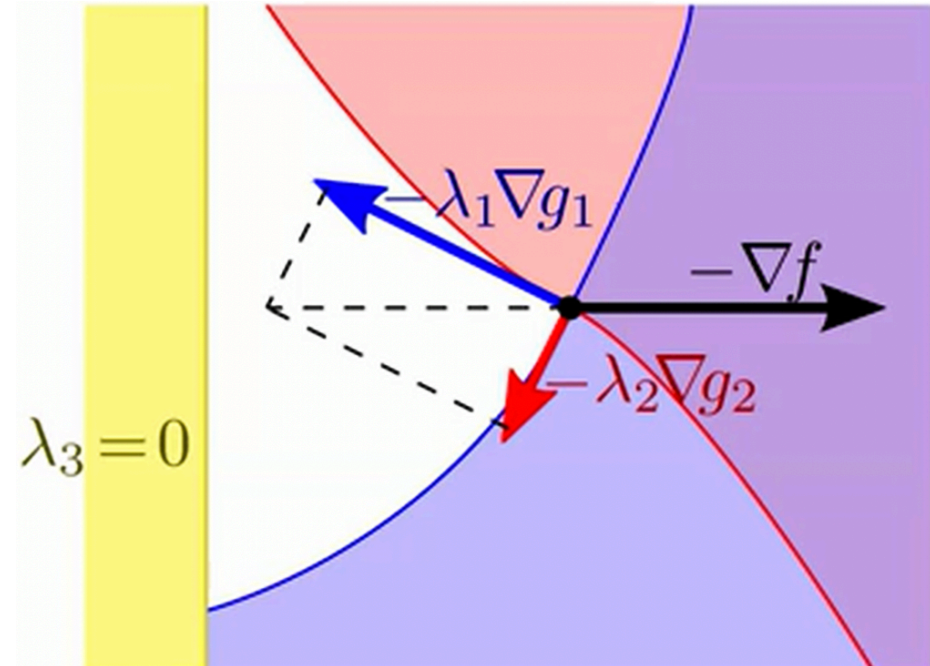
Upper elbow limit relaxed from 60° to 61° :

quantity	value
multiplier	$0.01192 \text{ m}^2/\text{rad}$
predicted cost change	$-2.08 \times 10^{-4} \text{ m}^2$
computed cost change	$-2.05 \times 10^{-4} \text{ m}^2$

Which fences contribute to the balance?

- Several touched fences can contribute to one balance
- An inactive fence has zero multiplier
- A touched fence may also have zero multiplier
- Equality multipliers can have either sign

We generally do not know the active constraints before solving.



One function collects the gradient balance

$$L(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\kappa}) = f(\mathbf{z}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{z}) + \boldsymbol{\kappa}^\top \mathbf{h}(\mathbf{z})$$

This is the **Lagrangian**. Bold constraint and multiplier symbols stack their scalar components.

$$\nabla_{\mathbf{z}} L = \nabla f + \sum_i \lambda_i \nabla g_i + \sum_j \kappa_j \nabla h_j$$

- At a KKT point, this gradient is zero
- Feasibility and the other KKT checks still matter
- The second half shows how solvers handle constraints and obtain multiplier estimates

The same requirements appear throughout robotics

robotics task	constrained formulation
choose a configuration	inverse kinematics with joint limits, clearance, and an exact goal
choose a motion	optimize the path with requirements at its sampled configurations
update a controller	choose a local motion while respecting limits
refine a learned proposal	check or improve the proposed configuration or motion with a constrained solver

A solver result still needs feasibility checks. Local convergence does not establish a globally best answer or physical safety.

Summary of 4.1

- **Feasibility:** obey every fence and rail
- **Balance:** the objective gradient can be nonzero when constraints prevent improvement
- **Activity:** inactive fences have zero multipliers; rail multipliers may have either sign
- **Sensitivity:** a multiplier predicts the local benefit of relaxing a requirement

The clipped elbow was allowed, but its shoulder was unfinished.

Next: how does a solver find a constrained answer?

Break

Second half: inverse kinematics as one nonlinear program, three ways a solver builds a fence out of costs, and whole trajectories solved the same way.

Inverse kinematics as a nonlinear program

$$\min_{\mathbf{q}} \frac{1}{2} \|\mathbf{q} - \mathbf{q}_{\text{ref}}\|_{\mathbf{D}}^2 \quad \text{s.t.} \quad \varphi(\mathbf{q}) - \mathbf{y}^* = \mathbf{0}, \quad \mathbf{q}^{\min} \leq \mathbf{q} \leq \mathbf{q}^{\max}, \quad r_o - \|\mathbf{p}_p(\mathbf{q}) - \mathbf{c}\| \leq 0$$

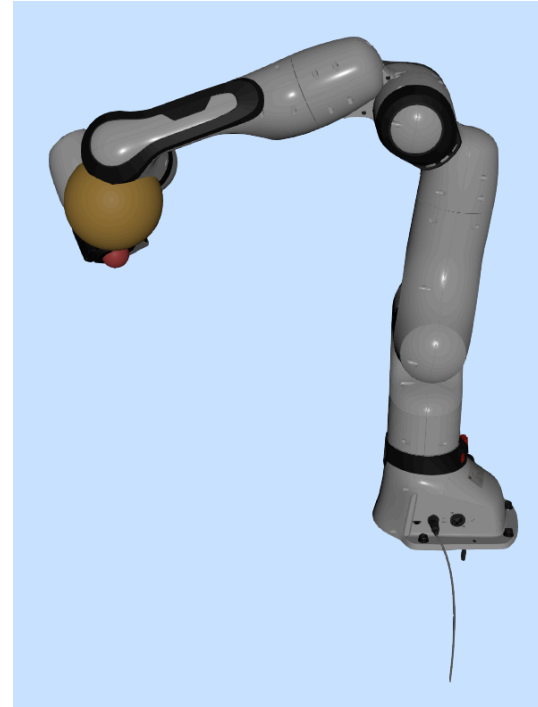
- Two changes from Deck 04:
 - target: soft residual → **hard equality**
 - cost: a **fixed** reference posture $\mathbf{q}_{\text{ref}}$ replaces the step penalty
- $\mathbf{q}_{\text{ref}}$ elsewhere: $\mathbf{q}_0$ in the essentials' §5, $\mathbf{q}_{\text{ready}}$ in the Panda demo
- With the elbow limited to $\pm 60^\circ$:
 - reach: 0.608 to 0.700 m from the base
 - target: 0.539 m away, so an exact target is **infeasible**
- Same slot, more features: gaze, feet stay put, balance, contacts (Deck 06, A2)

Live demo: IK as a nonlinear program on the Panda

Slide 17's program on a 7-joint arm; one solver, SNOPT, through Drake's `InverseKinematics`

term	on the Panda
cost	$\frac{1}{2} \ \mathbf{q} - \mathbf{q}_{\text{ready}}\ ^2$
rail	flange on the target: $\varphi(\mathbf{q}) - \mathbf{y}^* = \mathbf{0}$
fences	URDF joint limits; three wrist points clear of a sphere $(\mathbf{c}, r_o)$

live start	value
target $\mathbf{y}^*$	(0.45, 0.10, 0.45) m
sphere center $\mathbf{c}$	(0.45, 0.10, 0.53) m, 8 cm above the target
radius r_o	0.07 m



- **Predict:** the printed active set; its change 10 and 15 cm lower; a target inside the sphere
- Live: three sliders; the console prints cost and active set; a failed solve keeps the arm, greys the target

drake/03-ik-as-nlp/ik_nlp_panda.py --meshcat

▶ run it live

Panda results: active sets, multipliers and prices

run	target $\mathbf{y}^*$ (m)	active inequalities	λ	κ (target x, y, z)	cost
1	(0.45, 0.10, 0.45), no sphere	none	—	(−2.07, −0.67, 0.71)	0.209
2	(−0.20, 0, 0.60), behind the robot	joint 2 at its lower limit −1.7628 rad	1.053	(2.10, 4.28, 6.11)	1.499
3	run 1 + the sphere (the live start)	wrist point at 0.0700 m from $\mathbf{c}$	7.30	(2.63, −5.76, −0.02)	0.386

price check	shift	predicted	computed
run 1, target x	+1 mm	$-\kappa_x \times 0.001 = +0.00207$	+0.00208
run 2, joint-2 limit	+0.001 rad	$\lambda \times 0.001 = +0.00105$	+0.00105

- The rail is imposed in every run; its κ is nonzero here, but need not be
- Run 2 (print mode): joint 1 starts at +0.01 rad, ends at +0.856; the mirror (−0.856) costs the same, κ_y flips sign
- Optional: from the exactly symmetric $\mathbf{q}_{\text{ready}}$, SNOPT stops at a constrained saddle, **not a minimum** (cost 1.729; notes §5 (k))
- "No feasible point found" is a solver failure, **not** a proof; inside the sphere, infeasible by construction

How does a solver handle the fences? Three ways to build one out of costs

construction	augmented objective	outer loop	character
squared penalty	$S = f + \mu \sum_i [g_i > 0] g_i^2 + \mu \sum_j h_j^2$	$\mu \rightarrow \infty$	a ramp: trespasses wherever a fence must push ($\lambda^* > 0$)
log barrier	$B = f - \mu \sum_i \log(-g_i)$	$\mu \rightarrow 0$	a wall inside the fences: never touches, needs a feasible start; encodes fences only
augmented Lagrangian	$A = f + \sum_i [g_i > 0 \vee \lambda_i > 0] (\lambda_i g_i + \mu g_i^2) + \sum_j (\kappa_j h_j + \mu h_j^2)$	λ, κ updated between inner solves, μ fixed	a ramp plus a learned wind

Each inner problem: **unconstrained**, solved by Deck 04's Newton or Gauss-Newton (quasi-Newton BFGS: Deck 04's backup slides).

Squared penalty: a ramp on the forbidden side

$$S(\mathbf{z}; \mu) = f(\mathbf{z}) + \mu [g > 0] g(\mathbf{z})^2$$

1. Inner minimum: $\nabla f + 2\mu g \nabla g = \mathbf{0}$

2. So $2\mu g$ plays λ , and the violation is

$$g(\mathbf{z}') \approx \frac{\lambda^*}{2\mu}$$

- **Trespass wherever the fence must push** ($\lambda^* > 0$), shrinking like $1/\mu$
- An idle fence is not violated
- Hessian gains an eigenvalue of order μ : conditioning worsens as the answer improves

Essentials §5: the posture-weighted cost with a fixed $\mathbf{q}_{\text{ref}}$ is this penalty on the target rail, $\mu = K/2$

Elbow limit, from $(30^\circ, 45^\circ)$:

μ	q_2 (deg)	violation	$2\mu g$
1	60.338	0.338°	0.01181
10	60.034	0.034°	0.01191
100	60.003	0.003°	0.01192
1000	60.000	0.0003°	0.01192

$$\lambda^* = 0.01192$$

μ is chosen; λ is solved for (by hand on slides 6 and 10, by the outer loop on slide 23)

Log barrier: a wall that is never touched

$$B(\mathbf{z}; \mu) = f(\mathbf{z}) - \mu \log(-g(\mathbf{z}))$$

(1) stationarity of B : $\nabla f - \frac{\mu}{g} \nabla g = \mathbf{0}$

(2) name the push: $\lambda := -\mu/g > 0$

(3) substitute: $\nabla f + \lambda \nabla g = \mathbf{0}, \quad \lambda g = -\mu$

Complementarity relaxed by μ

Limits:

- strictly feasible start only
- added cost encodes fences only: a rail needs an equality-constrained inner solve (interior-point codes do that)
- Hessian term $\lambda^2 / \mu \nabla g \nabla g^\top$ grows as $\mu \rightarrow 0$ (active fence)

Elbow limit, from $(30^\circ, 45^\circ)$:

μ	q_2 (deg)	slack	$-\mu/g$
10^{-2}	28.833	31.2°	0.01838
10^{-3}	55.786	4.2°	0.01360
10^{-4}	59.527	0.47°	0.01212
10^{-6}	59.995	0.005°	0.01192

- every iterate strictly inside
- **Central path:** the minima $\mathbf{z}'(\mu)$ trace it to the optimum as $\mu \rightarrow 0$

Augmented Lagrangian: a ramp plus a learned wind

$$A = f + \lambda g + \mu g^2 \quad (\text{while } g > 0 \text{ or } \lambda > 0)$$

1. **Inner**, multipliers frozen: minimize A over $\mathbf{z}$; at its minimum:

$$0 = \nabla f + (\lambda + 2\mu g) \nabla g$$

the parenthesis: the new estimate of λ^*

2. **Outer**, $\mathbf{z}'$ frozen: update the multipliers

$$\lambda \leftarrow \max(\lambda + 2\mu g(\mathbf{z}'), 0), \quad \kappa \leftarrow \kappa + 2\mu h(\mathbf{z}')$$

The inner problem is least squares

Complete the square; the dropped constant does not depend on $\mathbf{z}$:

$$\lambda g + \mu g^2 = \frac{1}{2} \left(\sqrt{2\mu} \left(g + \frac{\lambda}{2\mu} \right) \right)^2 - \frac{\lambda^2}{4\mu}$$

Inner cost: $\frac{1}{2} \|\mathbf{r}\|^2$ plus a constant, with rows

residual row	value	Jacobian row
fence with $g > 0$ or $\lambda > 0$	$\sqrt{2\mu} \left(g + \frac{\lambda}{2\mu} \right)$	$\sqrt{2\mu} \nabla g^\top$
rail	$\sqrt{2\mu} \left(h + \frac{\kappa}{2\mu} \right)$	$\sqrt{2\mu} \nabla h^\top$

so **Deck 04's damped step solves it** (λ, κ fixed)

The augmented Lagrangian learns the multiplier

Theory: a fixed μ suffices

- above a problem-dependent threshold
- a local result

	μ	stop
our code	fixed	three residuals (slide 25)
RAI's <code>NLP_Solver</code> default (documented; installed growth option read)	$\times 5$ per outer step	step tolerance 0.01

Elbow limit, $\mu = 1$ fixed:

outer	λ before	q_2 (deg)	g	λ after
0	0	60.338	$+0.338^\circ$	0.011805
1	0.011805	60.003	$+0.003^\circ$	0.011917
2	0.011917	60.000	$+0.00003^\circ$	0.011918
3	0.011918	60.000	0	0.011918

lower limit: $\lambda_2 = 0$ throughout

Stopping with fences: three residuals, not one gradient

residual (updated multipliers)	meaning	elbow limit, AL $\mu = 1$, outer 3	KOMO solution A, outer 5
feasibility: $\max g^+$	on the allowed side	4.9×10^{-9}	0
feasibility: $\max h_j $	on every rail	— (no rail)	5.1×10^{-10}
Lagrangian stationarity: $\ \nabla f + \sum_i \lambda_i \nabla g_i + \sum_j \kappa_j \nabla h_j\ $	forces balance	1.3×10^{-11}	1.3×10^{-10}
complementarity: $\max_i \lambda_i g_i $	idle fences push nothing	5.9×10^{-11}	9.5×10^{-13}

$\|\nabla f\|$ alone: **not a sufficient test**

- elbow optimum: $\|\nabla f\| = 0.0119$
- KOMO path A: $\|\nabla f\| = 0.027$, stays there
- another constrained optimum: can be 0

Deck 04's words, new meanings:

status	meaning
solved	all three residuals small
stationary	force balance holds (current multipliers); another residual not settled
stalled	an inner solve rejects its direction
limit	outer iteration cap reached

Comparison

	squared penalty	log barrier	augmented Lagrangian
added cost encodes rails	yes	no : a rail needs an equality- constrained inner solve	yes
start anywhere	yes	no : strictly feasible start	yes
iterates feasible	no: trespass where $\lambda^* > 0$	yes, interior	no, until the multipliers settle
exact in the limit	$\mu \rightarrow \infty$	$\mu \rightarrow 0$	fixed μ above a threshold, as the multipliers converge (local)
conditioning	worsens as $\mu \rightarrow \infty$	worsens as $\mu \rightarrow 0$	μ need not grow for feasibility; the problem's own conditioning remains
where you meet it	soft costs everywhere	interior-point solvers (IPOPT)	KOMO; RAI's <code>NLP_Solver</code> (documented default $\mu \times 5$; installed growth option read)

The target is a constraint; posture is the cost

Slide 18's program without the sphere (slide 19's runs 1 and 2), solved by slide 24's loop with both weights fixed:

$$\min_{\mathbf{q}} \frac{1}{2} \|\mathbf{q} - \mathbf{q}_{\text{ready}}\|^2 \quad \text{s.t.} \quad \underbrace{\varphi(\mathbf{q}) - \mathbf{y}^* = \mathbf{0}}_{\text{rail } \mathbf{h}, \boldsymbol{\kappa}, \mu_h=1000 \text{ rad}^2/\text{m}^2}, \quad \underbrace{\mathbf{q}^{\min} \leq \mathbf{q} \leq \mathbf{q}^{\max}}_{14 \text{ fences, } \boldsymbol{\lambda}, \mu_g=100}$$

run	target $\mathbf{y}^*$ (m)	joint bounds at the answer	$\boldsymbol{\lambda}$
A (slide 19's run 1)	(0.45, 0.10, 0.45)	all 14 inactive	0
B (slide 19's run 2)	(−0.20, 0, 0.60)	joint 2 active at its lower bound −1.7628 rad	1.053 (joint 2 only)

Both answers are local, from the start shown on the page

- Slide 24: the squared distance was the **cost**. Here the distance is the rail's **violation** $\|\mathbf{h}\|$, driven to 0 within tolerance
- Inner step, a joint update: the arm moves, $\boldsymbol{\kappa}$ and $\boldsymbol{\lambda}$ frozen
- Outer step, a multiplier update, the arm fixed: $\boldsymbol{\kappa} \leftarrow \boldsymbol{\kappa} + 2\mu_h \mathbf{h}$ and $\lambda_i \leftarrow \max(0, \lambda_i + 2\mu_g g_i)$

From one configuration to a path

- Knots $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_T$; $\mathbf{q}_0$ fixed: the current configuration
- Costs on **differences**, today $\Delta t = 1$:
 - velocity $(\mathbf{q}_t - \mathbf{q}_{t-1})/\Delta t$
 - acceleration $(\mathbf{q}_{t-2} - 2\mathbf{q}_{t-1} + \mathbf{q}_t)/\Delta t^2$
- Constraints **at time slices**:
 - goal and rest: $t = T$
 - clearance: every t

One nonlinear program, nT variables.

$$\min_{\mathbf{q}_{1:T}} \sum_t \frac{1}{2} \|\mathbf{r}_t(\mathbf{q}_{t-k:t})\|^2$$

$$\text{s.t. } \mathbf{g}_t(\mathbf{q}_{t-k:t}) \leq \mathbf{0}, \quad \mathbf{h}_t(\mathbf{q}_{t-k:t}) = \mathbf{0}$$

decision variable:

$$\mathbf{z} = (\mathbf{q}_1, \dots, \mathbf{q}_T) \in \mathbb{R}^{nT}$$

$T = 1$, no rest constraint: slide 17's inverse kinematics, with

- $\mathbf{q}_{\text{ref}} = \mathbf{q}_0$
- $\mathbf{D} = \mathbf{I}$

The k-order Markov structure (KOMO)

- A feature uses at most $k + 1$ consecutive knots; k is the **Markov order**

k	features
0	position, distance, orientation
1	velocity
2	acceleration

$$\mathbf{r}_t = \mathbf{q}_{t-2} - 2\mathbf{q}_{t-1} + \mathbf{q}_t, \quad t = 1..T$$

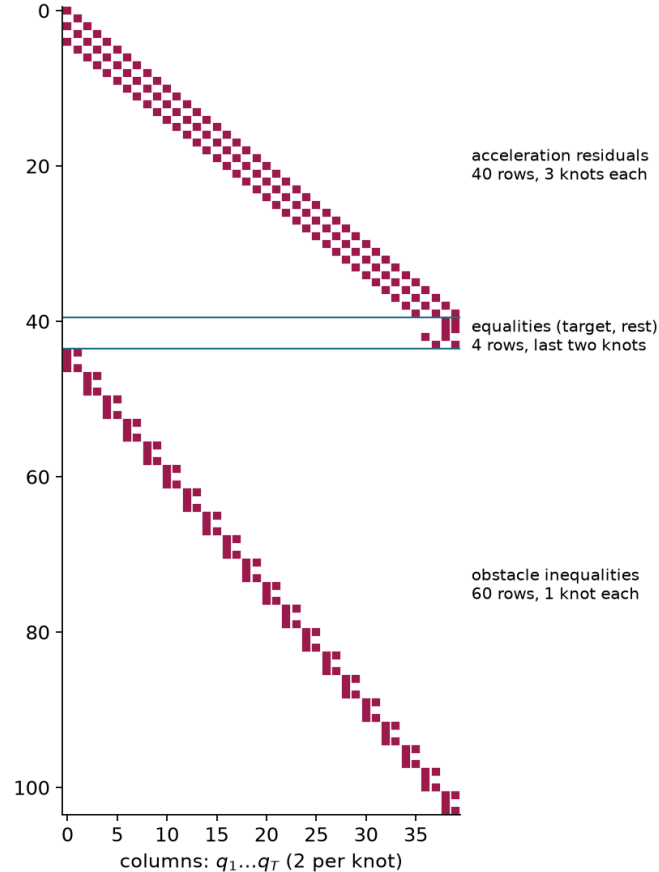
$$\mathbf{h} = (\varphi(\mathbf{q}_T) - \mathbf{y}^*, \mathbf{q}_T - \mathbf{q}_{T-1})$$

$$g_{t,p} = r_o - \|\mathbf{p}_p(\mathbf{q}_t) - \mathbf{c}\|$$

- Start from rest: $\mathbf{q}_{-1} := \mathbf{q}_0$, so $\mathbf{r}_1 = \mathbf{q}_1 - \mathbf{q}_0$
- A velocity cost jumps to full speed in one step; $k = 2$ penalizes that jump
- **k-order Markov Optimization (KOMO)** names this formulation; RAI's `k0rder=2` selects this order

Why k -order matters: every row is local in time

full feature Jacobian, 104 x 40: sparse row blocks, each row touches at most 3 knots, 5 % nonzero; the normal matrix $J^T J$ is block-banded



Left: full feature Jacobian

- size 104×40
- 5.3 % nonzero

Rows:

- each touches at most $k + 1$ nearby knots
- goal rows' block: Deck 03's $\mathbf{J}(\mathbf{q}_T)$
- so $\mathbf{J}_r^T \mathbf{J}_r$ is **block-banded**

Operation count of one step (an estimate, not a timing):

$$O(T (k + 1)^2 n^3) \text{ instead of } O((nT)^3)$$

T	k	n	banded	dense
100	2	7	$\approx 3 \times 10^5$	$\approx 3 \times 10^8$

Two Jacobians, both sparse

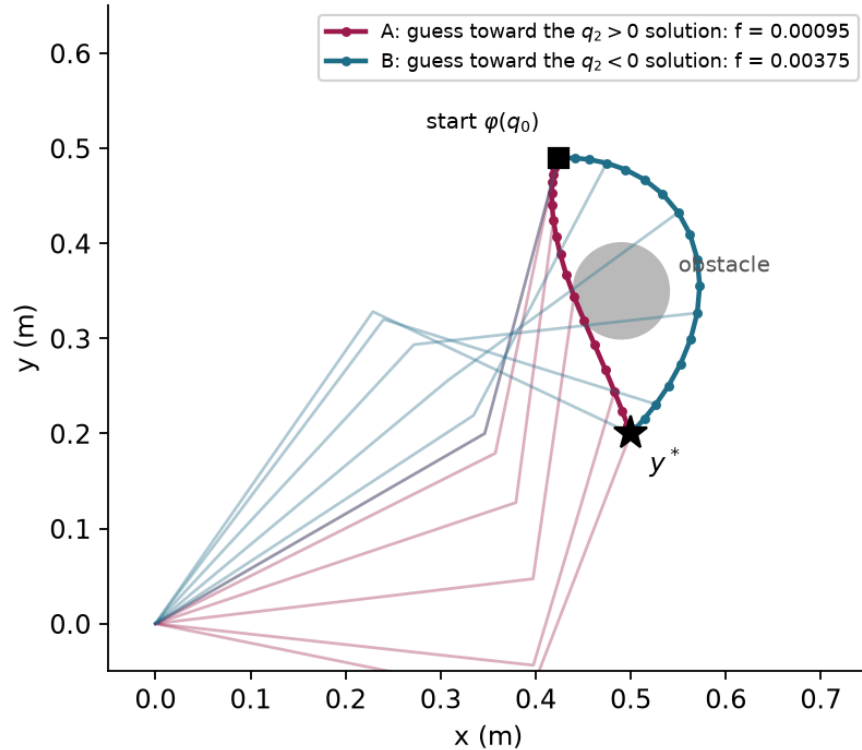
- **Full feature Jacobian:** every feature row, unscaled (slide 30's figure)
- **Inner residual Jacobian:** the rows that slide 23's inner solve factorizes

rows	full feature Jacobian	inner residual Jacobian, solution A
smoothness $\mathbf{r}_t$	40	40
rails $\mathbf{h}$	4	4, times $\sqrt{2\mu}$
fences $\mathbf{g}$	60	2 (idle ones dropped), times $\sqrt{2\mu}$
size	104×40	46×40
nonzero	5.3 %	—

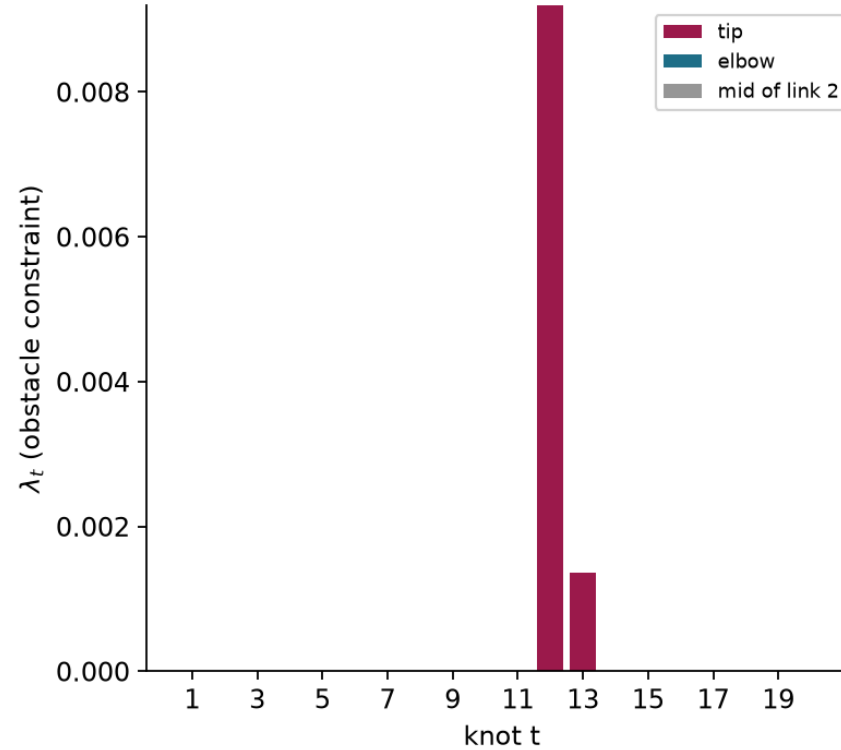
- Smoothness rows alone, $T = 20$: condition number 2.5×10^5
 - grows like T^4 (Deck 04 slide 9)

Worked example (i): KOMO on the two-link arm

tip paths of the two local optima ($T = 20, k = 2$), arm drawn every 4th knot



solution A: the fence pushes only at knots 12 and 13



T	k	start q_0	target y^*	disc r_o	points per knot	solver	μ
20	2	$(30^\circ, 45^\circ)$	$(0.5, 0.2)$	5 cm	3	slide 23's augmented Lagrangian	10, fixed

Right: λ_t , the fence's push, only at knots 12–13. No elbow limit here: under $\pm 60^\circ$ the target is out of reach (slide 10)

Two guesses, two valleys

	guess A	guess B
initial guess: joint-space line to	the elbow $+80.4^\circ$ solution	the elbow -80.4° solution
trespass before iteration 1	4.8 cm	none
ours (augmented Lagrangian): path	below the disc	above the disc
ours: f	0.000946	0.003746
ours: contact	knots 12–13	none
Drake + SNOPT: f	0.000957, a <i>different</i> KKT point	same path to 0.003° , same f
Drake + SNOPT: contact	knots 11–12	none
Drake + SNOPT: KKT residual	about 4×10^{-7}	—

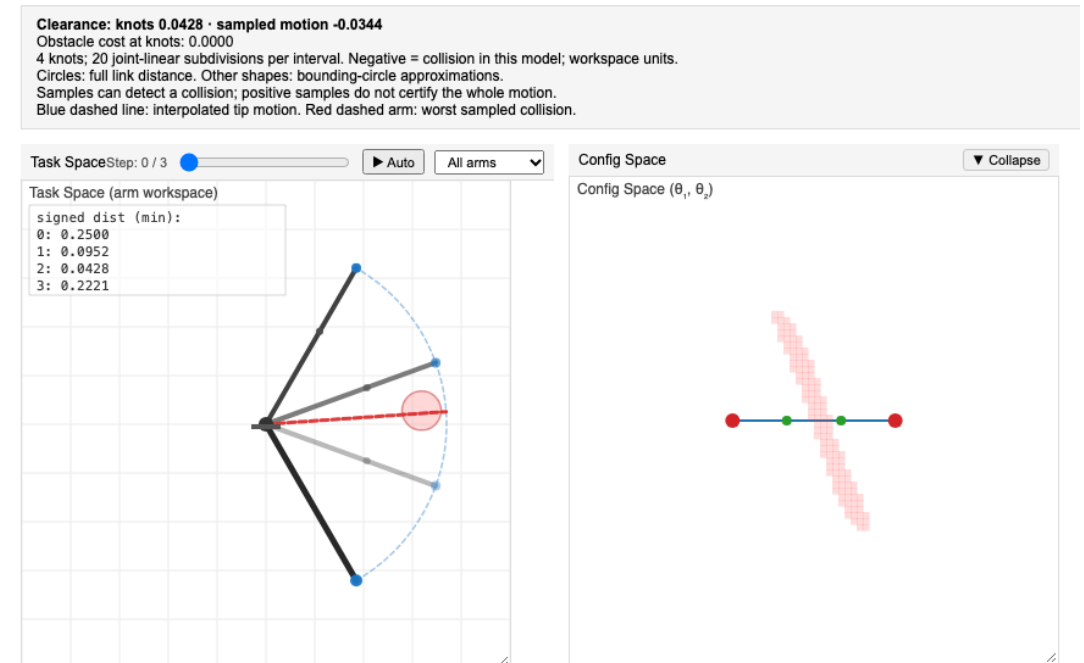
- Ours, restarted at SNOPT's point A: does not move
- Drake's rail duals = $-\kappa$ (Drake's sign)

Fog: the guess decides the valley. **Report the guess with the answer.**

Live demo: clear knots can hide a collision

- Opens on the **Trajectory** tab with **Between-knot collision** loaded: four arm poses, all clear; obstacle cost **0**
- **Predict:** is the motion between those poses clear too?
- Interpolate the joints: the **red dashed arm collides**
 - knot clearance: positive
 - sampled-motion clearance: negative
- Choose **Add midpoint knots**: same joint path, seven knots, now a **positive obstacle cost**

More knots reveal a missed collision. They do not repair the path or certify continuous clearance.



KOMO as a Drake program, line by line

```
# code/komo_two_link.py --drake: slide 29's program, handed to SNOPT
prog = MathematicalProgram()
z = prog.NewContinuousVariables(N, "z") # z = (q_1, ..., q_T)
_, A = acc_residuals(np.zeros(N)); b, _ = acc_residuals(np.zeros(N)) # r = A z + b
prog.Add2NormSquaredCost(np.sqrt(0.5)*A, -np.sqrt(0.5)*b, z) # cost  $\frac{1}{2}\|A z + b\|^2$ 
bh = prog.AddConstraint(h_fn, np.zeros(4), np.zeros(4), z) # rails
bg = prog.AddConstraint(g_fn, -np.inf*np.ones(3*T), np.zeros(3*T), z) # fences
prog.SetInitialGuess(z, z_init); result = SnoptSolver().Solve(prog)
zd = result.GetSolution(z); lam = result.GetDualSolution(bg) # path, fence duals
```

Slide 29's three kinds of term, nothing else:

term	Drake call	lb	ub
cost	Add2NormSquaredCost (an AddCost)	—	—
rail	AddConstraint	0	0
fence	AddConstraint	$-\infty$	0

- Guess A: SNOPT ends at a **second KKT point** of the same program (costs on slide 33)
 - knots up to 2.9° apart from ours
 - slide 23's loop restarted there stays within 10^{-4} degrees
- Guess B: both solvers agree to 0.003°

Failure modes (1): posing and stopping

- **Sign of g :** "distance at least r_o " is $r_o - \|\mathbf{p}_p - \mathbf{c}\| \leq 0$; reversed, the obstacle's inside is feasible
- **Negative λ :** fails dual feasibility, so not a KKT point as posed; a negative κ is fine
- **Penalty "converged" but violated:** a finite μ trespasses wherever the fence must push; check g , not only the step norm
- **Stopping on $\|\nabla f\|$ with fences:** 0.012 at the elbow optimum; use slide 25's three residuals

Failure modes (2): solvers and certificates

- **Barrier from outside, or on a rail:** undefined; too large a μ leaves Newton at a saddle point
- **One solver's answer is not "the optimum":** two KKT points 1 % apart from one guess; report the guess and the residuals
- **"Infeasible" from a solver:** a local failure to find a feasible point, not a proof; the elbow case is proved by reach (slide 17)
- **Constraints hold at the knots only:** 2.07 mm trespass between knots 12–13 at $T = 20$, joint-space interpolation; more knots reveal it, none certify clearance

Summary; quiz 2; A1; the next deck

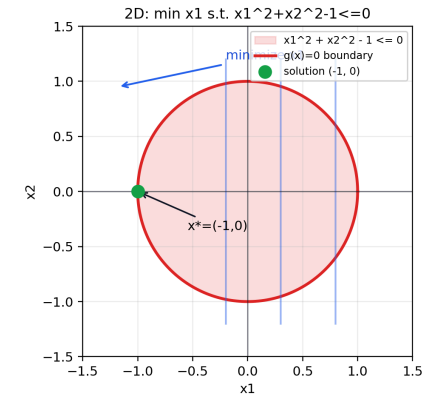
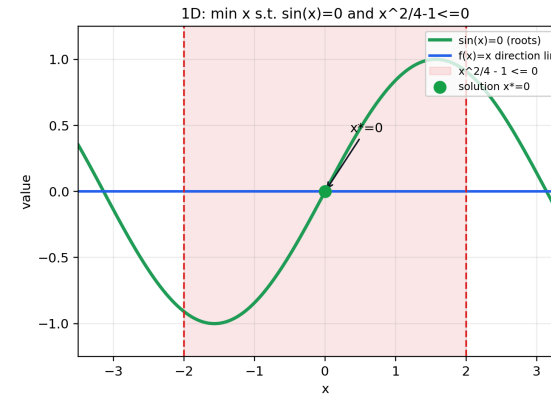
- **4.1:** fences and rails; force balance $\nabla f + \lambda \nabla g = \mathbf{0}$; a fence pushes only if touched (complementarity); λ is a price
- **4.2:** inverse kinematics and trajectories as one nonlinear program
 - k -order features: Jacobian rows local in time, so $\mathbf{J}^\top \mathbf{J}$ is block-banded
 - fences into costs: penalty, barrier, augmented Lagrangian (RAI's documented default; option values checked)
 - done when feasibility, force balance and complementarity are all small
- **R 5:** quiz 2 (Gauss-Newton, KKT, KOMO, control stack, RRT); **A1 due**
- **Deck 06 (4.3–4.4):** the control stack, PD and MPC in one slide; configuration space and sampling-based planning

Backup slides

Backup · Solve by sketching

1. $\min x$ s.t. $\sin x = 0$, $x^2/4 - 1 \leq 0$: which constraints are active, which multipliers nonzero?
2. $\min x_1$ s.t. $x_1^2 + x_2^2 - 1 \leq 0$: force balance at the solution?
3. Wedge: $\min x + y$ s.t. $0.5x - y - 0.2 \leq 0$, $-x + 0.5y - 0.2 \leq 0$
 - solution: $(x, y) = (-0.4, -0.4)$, both active
 - multipliers: $\lambda_0 = \lambda_1 = 2$

Constrained Optimization Sketches (Session 5)



Backup · The price of a constraint, derived

Setup, perturbed problem: $\min f(\mathbf{z})$ s.t. $g(\mathbf{z}) \leq b$, active near $b = 0$

- solution $\mathbf{z}^*(b)$, optimal cost $f^*(b) = f(\mathbf{z}^*(b))$
- assumption: the same fences stay active for small b , so $\mathbf{z}^*(b)$ moves smoothly

1. Differentiate the identity $g(\mathbf{z}^*(b)) = b$: $\nabla g(\mathbf{z}^*)^\top \frac{d\mathbf{z}^*}{db} = 1$

2. Differentiate the cost; use stationarity $\nabla f = -\lambda \nabla g$:

$$\frac{df^*}{db} = \nabla f(\mathbf{z}^*)^\top \frac{d\mathbf{z}^*}{db} = -\lambda \nabla g(\mathbf{z}^*)^\top \frac{d\mathbf{z}^*}{db} = -\lambda$$

3. Several constraints: $\partial f^* / \partial b_i = -\lambda_i$; inactive ones have $\lambda_i = 0$ and no price

Backup · The central path, and the barrier's relaxed KKT

At the inner optimum of $B(\mathbf{z}; \mu)$, with

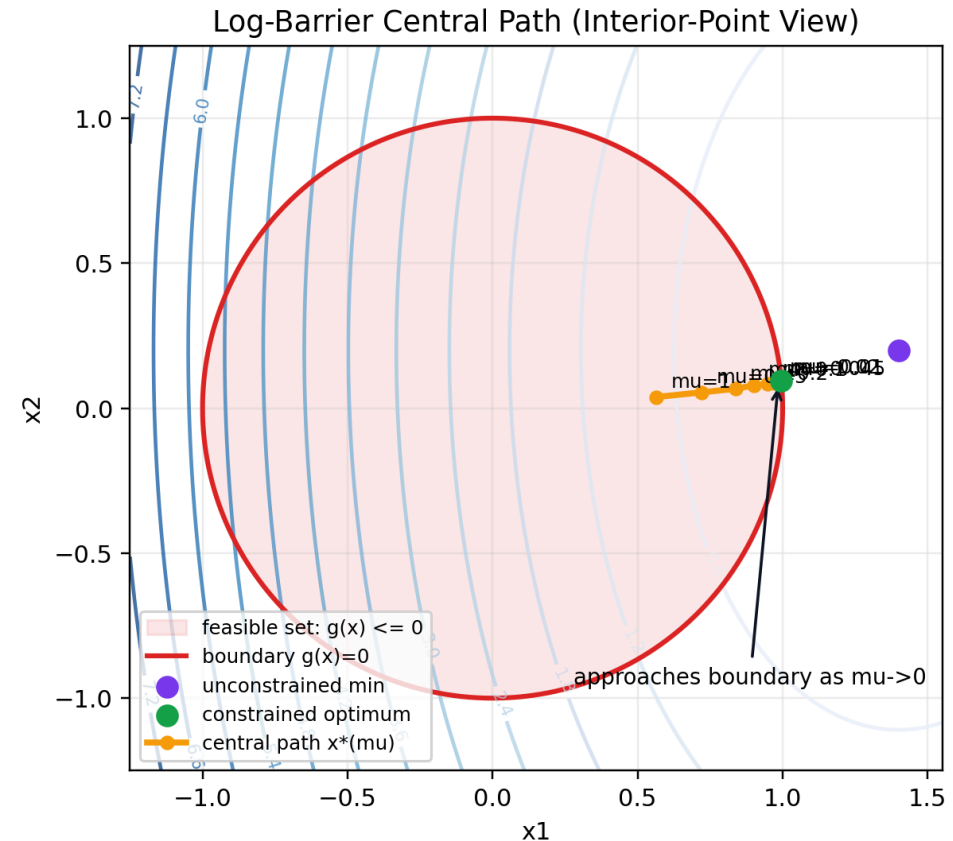
$$\lambda_i := -\mu/g_i > 0$$

condition	exact KKT	barrier optimum
stationarity	$\nabla f + \sum \lambda_i \nabla g_i = \mathbf{0}$	same
dual feasibility	$\lambda_i \geq 0$	automatic
primal feasibility	$g_i \leq 0$	strict, $g_i < 0$
complementarity	$\lambda_i g_i = 0$	$\lambda_i g_i = -\mu$

Duality gap bound (**convex** only, m fences):

$$f(\mathbf{z}'(\mu)) - f^* \leq m\mu$$

- arm's saddle run (notes §7 (6)): fails by four orders of magnitude



Backup · Gradient and Hessian of the barrier

$$\nabla_{\mathbf{z}} B = \nabla f - \mu \sum_i \frac{1}{g_i} \nabla g_i, \quad \nabla_{\mathbf{z}}^2 B \approx \nabla^2 f + \mu \sum_i \frac{1}{g_i^2} \nabla g_i \nabla g_i^\top$$

- $-\mu/g_i > 0$: a repulsive force, growing as the fence is approached
- Hessian model drops $\nabla^2 g_i$, as Gauss-Newton drops the residuals' second derivatives
- Added term: positive semidefinite, size λ_i^2/μ
 - active fence: unbounded as $\mu \rightarrow 0$; conditioning worsens, like the penalty's
- Inner solver: Newton on B with this model, Armijo backtracking, warm start
 - stop when $m\mu$ is below the tolerance: a valid test for convex problems only

Backup · The augmented Lagrangian, derived

1. One rail: $A = f + \kappa h + \mu h^2$; stationarity at the inner minimum z' :

$$0 = \nabla f + \kappa \nabla h + 2\mu h \nabla h = \nabla f + (\kappa + 2\mu h) \nabla h$$

2. Match with KKT, $\nabla f + \kappa^* \nabla h = 0$: the parenthesis is the new κ

3. Linear f and h : exact after one update

One-dimensional example (table at right):

- problem: $\min \frac{1}{2} z^2$ s.t. $z - 1 = 0$
- penalty weight: $\mu = 1$
- inner minimum: $z' = (2 - \kappa)/3$

outer	κ	z'	$h(z')$	κ new
0	0	0.6667	-0.3333	-0.6667
1	-0.6667	0.8889	-0.1111	-0.8889
2	-0.8889	0.9630	-0.0370	-0.9630
3	-0.9630	0.9877	-0.0123	-0.9877
∞	-1	1	0	-1

h shrinks by $1/(1 + 2\mu) = 1/3$ per outer iteration at **fixed** μ .

Backup · The full algorithm with inequalities

```
μ ← μ0; λ ← 0; κ ← 0
repeat
  z ← argmin_z f(z) + Σ_i [g_i(z) > 0 ∨ λ_i > 0] (λ_i g_i(z) + μ g_i(z)^2) + Σ_j (κ_j h_j(z) + μ h_j(z)^2) # inner: Newton / Gauss-Newton / BFGS, warm-started
  λ_i ← max(λ_i + 2μ g_i(z), 0) # fences: clamp at zero
  κ_j ← κ_j + 2μ h_j(z) # rails
  optionally μ ← c μ # RAI's documented default: c = 5
until max_i g_i(z)+, max_j |h_j(z)| ≤ 10-9 # feasibility (slide 25's three residuals,
and ||∇f + Σ_i λ_i ∇g_i + Σ_j κ_j ∇h_j|| ≤ 10-8 # Lagrangian stationarity with the updated multipliers;
and max_i |λ_i g_i(z)| ≤ 10-11 # complementarity stop also if an inner solve stalls)
```

- The bracket $[g_i > 0 \vee \lambda_i > 0]$ decides activity **pointwise**, at every evaluation: no enumeration of active sets
- $\lambda_i > 0$ with $g_i < 0$: a transient after a constraint stops being active; the clamp drives it to zero
- Theory: a fixed μ above a problem-dependent threshold suffices
- Practice: growth speeds convergence (RAI's documented default: slide 24)

Backup · The solver landscape, and the saddle-point view

method	idea
augmented Lagrangian	guess the multipliers, minimize, correct them from the residual
primal-dual Newton	Newton on the KKT system in $(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\kappa})$ jointly; interior-point solvers (IPOPT)
sequential quadratic programming	a quadratic program (QP) with the linearized constraints per step; multipliers inherited from the QP (SNOPT)
active-set	guess the touched fences, solve the equality problem, revise

