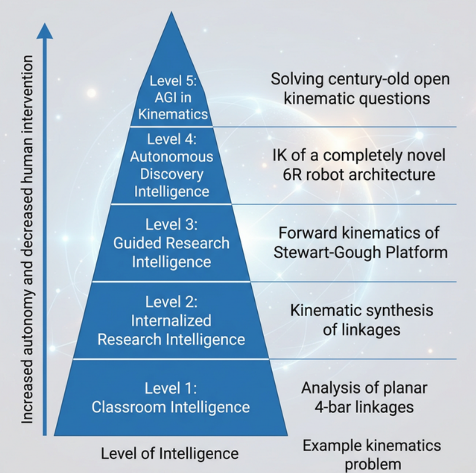
IDR Lab
Intelligent Design and Robotics Laboratory
GEN-AI DESIGN MR SIMULATION

The Ohio State University

Where Are We on the Path Toward AGI in Kinematics?



Benchmarking LLMs on Classical Kinematics Problems and the General 6R IK Challenge

Haijun Su, Professor
Intelligent Design and Robotics Laboratory
Department of Mechanical and Aerospace Engineering
The Ohio State University

ASME IDETC/CIE 2026
15-minute talk + 5-minute Q&A

The Coding-Agent Landscape Has Moved Fast

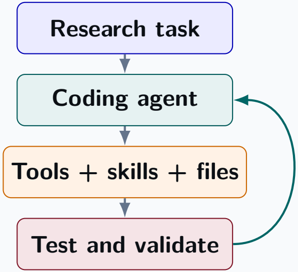
A dated benchmark snapshot

This work was produced in late 2025 and early 2026: an evidence-based snapshot of LLM performance on difficult kinematics problems with expert-specified routes and independent validation.

- **OpenAI Codex** and **Claude Code** couple models to a **harness**: workspace, tools, shell, tests, and iterative edits.
- **Agent skills** package reusable instructions, tools, and examples—a manual for recurring coding tasks.
- **Harness, loop, and graph engineering** are emerging ways to organize tool use, verification, and multi-step execution.

Interpret the paper carefully

Stronger execution environments can improve implementation reliability; they do not prove autonomous selection of a new kinematic derivation.



Can the agent choose the right symbolic strategy, not merely execute it?

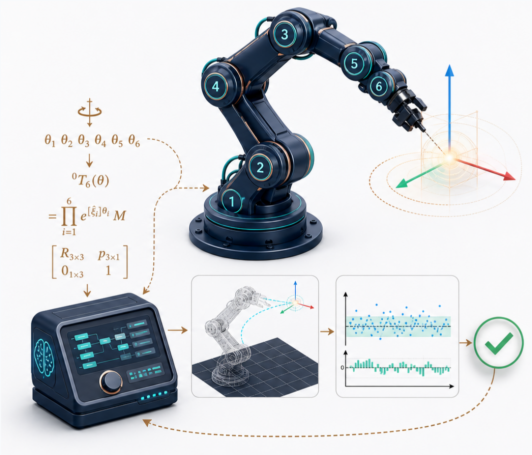
Goal: Measure What LLMs Can Actually Do in Kinematics

This paper's goal

Assess the capability of LLMs to generate **correct, executable, and independently validated** solvers for difficult classical kinematics problems.

- Benchmark coding-agent output on established problem families, anchored by **general 6R inverse kinematics**.
- Separate **implementation competence** from **symbolic-strategy discovery**.
- Count success only after executable code passes FK–IK–FK or equivalent independent checks.

The central question: can an agent synthesize and validate a solver—or only follow an expert-provided route?



A Domain-Specific AGI Benchmark

Paper's benchmark question

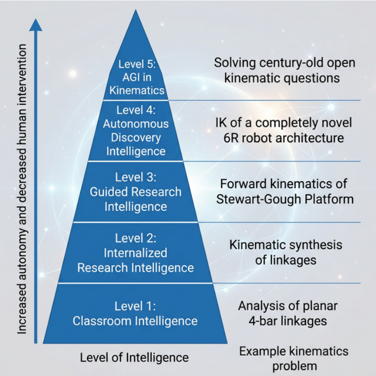
If all knowledge after 1980 were removed from training, could an AI independently derive the inverse kinematics of a general spatial 6R robot, or an equivalent 7R closed-loop solution framework?

- Inspired by the "Einstein test" framing for AGI.
- Passing it would indicate more than code generation skill; it would require original symbolic strategy selection.
- The paper argues that **symbolic reasoning** and **computational scale** should be evaluated separately.

What does *not* count?

- Throwing more compute at a known polynomial system
- Replaying a documented derivation without choosing the symbolic route
- Producing plausible code that fails FK–IK–FK verification

Five-Level Hierarchy for AI in Kinematics



- **Level 1** Classroom analysis
- **Level 2** Autonomous solution of established research tasks
- **Level 3** Complex reproduction using provided literature
- **Level 4** Solving solvable-but-undocumented problems
- **Level 5** Open-question AGI in kinematics

Working interpretation: the reported experiments cover Levels 1–3 directly and only touch the boundary of Level 4 under expert guidance.

Reported Benchmark Suite

Problem category	Level	Session time	Critical expert guidance
Planar and spherical linkages	1	< 5 min	None
Spatial SS dyad synthesis	2	~ 15 min	Eigenvalue method
Stewart–Gough FK	3	~ 30 min	Zero-dimensional ideal elimination
General 6R robot IK	3	Multi-hour to multi-day	14 equations + robust eigenvalue step
6R IK, special architectures	4	Multi-hour to multi-day	Decoupling and elimination order

- Difficulty increases with **symbolic bookkeeping and numerical strategy dependence**, not just polynomial degree.
- The most important contrast is Stewart–Gough FK vs. general 6R IK: both are hard, but only one required extensive corrective guidance.

Human–AI Workflow Used in the Paper

Three-phase workflow

1. Problem specification
2. Structured prompt design
3. Iterative refinement and validation

Prompt structure

- **Context**: role and references
- **Task**: target derivation/implementation
- **Requirements**: tolerances and tests
- **Deliverables**: scripts, solvers, report

Why this matters

- The final single-shot success on 6R IK is **not** raw autonomy.
- It is the result of expert prompt engineering that encodes the algebraic strategy and the acceptance tests.
- The paper treats validation as part of the benchmark, not an afterthought.

Operational lesson

LLMs are strongest when each symbolic stage is made explicit and independently checkable.

The General 6R Inverse-Kinematics Problem

Problem definition

Given $T^d \in SE(3)$, find all $\mathbf{q} = (\theta_1, \ldots, \theta_6)$ such that $T_1^6(\mathbf{q}) = T^d$. A **general** spatial 6R chain has no special simplifying axis arrangement.

Why it became a classical benchmark

- Raghavan–Roth (1993): dialytic elimination, a 16th-degree characteristic polynomial, and **at most 16** configurations.
- The challenge remains symbolic correctness plus stable recovery of every real pose.

Reference: Raghavan and Roth, *ASME Journal of Mechanical Design*, 115(2):314–320, 1993.

Why the General 6R IK Benchmark Matters

- The general 6R serial manipulator admits up to **16 real assembly configurations**.
- The solution requires a dense symbolic pipeline: kinematic split, bilinear constraint construction, elimination, lifting, and numerically stable root extraction.
- The Raghavan–Roth paper gives the analytical route, but not the most robust final numerical solve strategy.

14

scalar bilinear equations

12×12

dialytic lifted matrix

24×24

generalized eigenvalue pencil

Observed bottleneck

The hardest step was not writing syntax. It was choosing and enforcing the **correct symbolic route** and a **stable generalized eigenvalue formulation**.

Reference: Raghavan and Roth, *ASME Journal of Mechanical Design*, 115(2):314–320, 1993.

Kinematic Split Behind the 6R Solution

Rearranged forward kinematics

$$A_{3v} A_4' A_5 = A_2^{-1} A_1^{-1} A_{\text{hand}} A_6^{-1}$$

- $A_{3v} = R_z(\theta_3)$ isolates the variable part of joint 3.
- $A_4' \equiv A_{3s} A_4$ absorbs the structural constants from A_3.
- Left-hand side depends on $(\theta_3, \theta_4, \theta_5)$.
- Right-hand side depends on $(\theta_1, \theta_2, \theta_6)$.
- Extracting the position vector $\mathbf{p}$ and approach vector ℓ removes θ_6 from the key equations.
- This enables a bilinear separation between (θ_4, θ_5) and (θ_1, θ_2).

Fourteen Bilinear Constraints

Equation type	Count
Position $\mathbf{p}$	3
Approach ℓ	3
$\mathbf{p} \cdot \mathbf{p}$	1
$\mathbf{p} \cdot \ell$	1
$\mathbf{p} \times \ell$	3
$(\mathbf{p} \cdot \mathbf{p})\ell - 2(\mathbf{p} \cdot \ell)\mathbf{p}$	3
Total	**14**

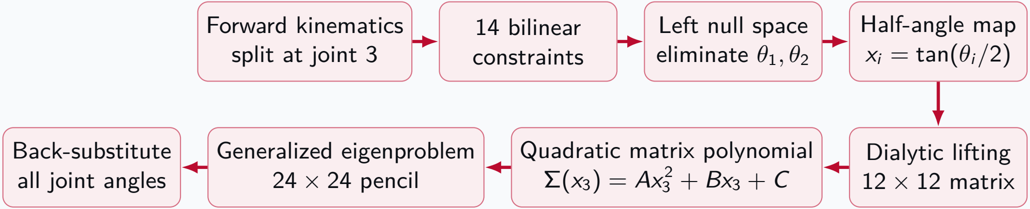
Shared linear structure

$$P_r\,\mathbf{m}_{45} = Q_r\,\mathbf{n}_{12}, \qquad r = 1, \ldots, 14$$

$$\mathbf{m}_{45} = [s_4 s_5,\ s_4 c_5,\ c_4 s_5,\ c_4 c_5,\ s_4,\ c_4,\ s_5,\ c_5,\ 1]^T$$

$$\mathbf{n}_{12} = [s_1 s_2,\ s_1 c_2,\ c_1 s_2,\ c_1 c_2,\ s_1,\ c_1,\ s_2,\ c_2]^T$$

Elimination Pipeline Used by the Solver



- The **Manocha–Canny generalized eigenvalue formulation** is the key numerical robustness upgrade relative to direct root-finding.
- The paper's main claim is that current LLMs need expert help at exactly these symbolic and numerical choke points.

References: Raghavan and Roth (1993); Manocha and Canny, *IEEE Transactions on Robotics and Automation*, 1994.

Where the Model Initially Failed

Failure mode 1

Equation extraction

Dense equations were misread without OCR, creating sign and indexing mistakes before implementation even began.

Failure mode 2

Symbolic route

The model produced mathematically plausible derivations that used the wrong monomial basis or an inconsistent elimination structure.

Failure mode 3

Numerical stability

Direct root-finding on the final characteristic polynomial was too unstable to recover all valid solutions reliably.

Human intervention that mattered

- enforce OCR-based reference reading
- specify the 14-equation construction exactly
- inject the generalized eigenvalue method

Validation Benchmark: The Manseur–Doty 16-Root Case

i	a_i	d_i	α_i (rad)	α_i (deg)
1	0.3	0.0	1.57	90.0
2	1.0	0.0	1.00	57.3
3	0.0	0.2	1.57	90.0
4	1.5	0.0	1.00	57.3
5	0.0	0.0	1.57	90.0
6	0.0	0.0	1.00	57.3

Reference planted joint tuple

$$\mathbf{q}^* = [0^\circ,\ 107.46^\circ,\ 112.46^\circ,\ -7.66^\circ,\ 0^\circ,\ 0^\circ]$$

16

expected real IK solutions

FK–IK–FK

validation cycle used to accept or reject solutions

10^{-9}

reported worst-case position residual threshold achieved on the benchmark

Recovered Accuracy on the 16-Solution Benchmark

16/16

solutions recovered

$< 10^{-9}$

all FK position residuals

8 pairs

shared (θ_1, θ_2) with symmetric wrist/elbow postures

Sol.	θ_1	θ_2	θ_3
4	0.00	107.46	112.46
5	-24.19	90.31	123.04
9	109.73	121.72	51.12
14	0.00	107.46	-67.54

Key observation

Solution 4 matches the planted reference tuple, and the full 16-solution set satisfies the manuscript's residual checks.

Runtime and Implementation Outcome

0.161 s
mean Python IK time

0.0015 s
mean C++ IK time

$106\times$
reported speedup

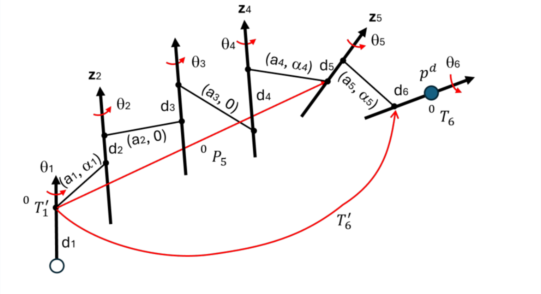
Coverage on 1000 random IK queries

- C++ solver preserved **999/1000** successful solves.
- The one miss occurred near a singular configuration.
- Reported cause: a spurious complex root filtered out by the real-root test.

Interpretation

The paper's contribution is not just symbolic reconstruction. It also shows that the generated pipeline can be turned into a practical high-throughput solver.

Example: 6R IK with a Parallel-Joint Architecture



Joint axes 2, 3, and 4 are parallel: $\alpha_2 = \alpha_3 = 0$.

A Level-4-boundary example

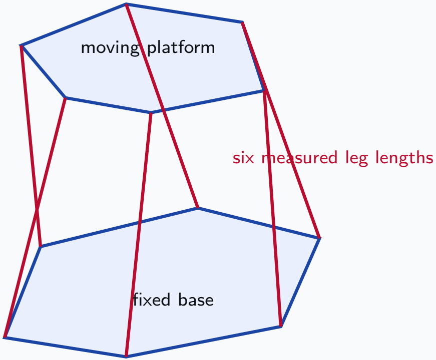
- Parallel axes change the algebraic structure: a bilinear problem in θ_1 and θ_6, then a planar 3R subproblem.
- No ready-to-use generic derivation was supplied for arbitrary DH parameters in this architecture class.
- Expert guidance supplied the decoupling and elimination order; the agent implemented and validated the pipeline.

100%

reported success across 119 industrial and 100 random parallel-joint robots

Reference: Su, *Mechanism and Machine Theory*, 222:106392, 2026. Figure: IDR Lab project page.

Companion Case: Forward Kinematics of a Stewart–Gough Platform



The FK task

Given six leg lengths and platform geometry, recover all physically valid poses of the moving platform.

40

maximum assembly configurations for a general platform

- A documented elimination cascade yields a 40th-degree characteristic polynomial.
- Level 3: the LLM reproduced that published sequence with exact arithmetic and a planted-root check.

References: Raghavan, *ASME Journal of Mechanical Design*, 1993; Husty, *Mechanism and Machine Theory*, 1996.

More Solved Problems: See the Companion IDETC 2026 Paper

Three further case-study families

- **Planar and spherical linkages**: one well-scoped prompt often sufficed.
- **Spatial SS dyad**: a companion-matrix generalized-eigenvalue reformulation recovered all 20 solutions.
- **Stewart–Gough FK**: exact-arithmetic elimination enabled a 40th-degree polynomial and planted-root verification.

Take the next step

For prompt templates, case-study detail, and validation evidence, see the **companion Paper 2 presentation**.

3

case-study families in Paper 2

20/20

SS-dyad solutions after expert numerical reformulation

40

degree of the Stewart–Gough characteristic polynomial

Reference: Su and McCarthy, *Solving Kinematics Problems by Leveraging the Power of Large Language Models*, ASME IDETC/CIE 2026 (Paper 2).

What Intelligence Level Do Current LLMs Reach?

- **Confirmed**: strong **Level 3** performance on documented, complex kinematics pipelines when references and structured prompts are provided.
- **Partial**: **approaching Level 4** under expert supervision for undocumented 6R special architectures.
- **Not shown**: autonomous Level 4 discovery or Level 5 AGI-level re-derivation of the general 6R method from scratch.

Paper's conclusion

Current state-of-the-art LLMs are powerful accelerators for analytical implementation, but they still depend on human experts for elimination strategy selection and numerically stable formulation choices.

Level 3

supported directly by reported results

Level 4

only in guided settings

Level 5

still a conceptual benchmark

Limitations and Reproducibility

Limitations

- Classical benchmark focus, not open problems
- Single-expert workflow
- Prompt wording and model-version sensitivity
- Verifier can reject bad solutions, but does not teach the right symbolic route

Why the authors still call it reproducible

- Final refined prompts are public in the companion repository.
- Acceptance criteria are objective: published benchmark behavior and FK–IK–FK residuals.
- The validated single-shot prompt reproduces the pipeline *after* expert prompt engineering.
- A reusable agent `SKILL.md` can package a problem family's method, tools, and validation checks; it was authored during the paper-era workflow.

Takeaways and Reproducibility Resources

Key conclusion

- LLMs can compress difficult analytical implementation into hours or minutes when the symbolic route is well specified.
- The general 6R IK benchmark exposes the boundary: expert reasoning remains decisive at elimination and numerical-stability bottlenecks.
- Progress toward AGI requires independent selection of valid symbolic strategies and validation without critical hints.

Practical implication

Near-term value is expert-amplifying, validation-driven code generation for difficult documented methods.

Prompts, generated code, and benchmark test reports

All materials are available through Prof. Su's public GitHub repositories:

- https://github.com/haijunsu-osu/IK_6R_RR_1993
- https://github.com/haijunsu-osu/llm_kinematics_deep_research_report
- https://github.com/haijunsu-osu/gdl_llm_reproduction
- https://github.com/haijunsu-osu/llm_FK_Stewart_Husty_1996
- https://github.com/haijunsu-osu/llm_ss_synthesis_liao_2001

Repositories provide the problem-specific prompts, generated scripts/code, and validation or benchmark reports described in the papers.

Questions

Thank you.

- The appendix includes the full benchmark summary, a larger slice of the 16-solution table, and the refined prompt structure.
- Contact: `su.298@osu.edu`

Appendix: Full Reported Benchmark Summary

Problem Category	Level	Time	Prompt Iter.	Critical Expert Guidance
Planar & spherical linkages	1	$<$ 5 min	Single-shot	None
Spatial SS dyad synthesis	2	$\sim$ 15 min	Few	Specifying eigenvalue method
Stewart–Gough FK	3	$\sim$ 30 min	Single-shot	Directing zero-dimensional ideal elimination
General 6R robot IK	3	Multi-hour to multi-day total	Many	14-equation derivation, generalized eigenvalue step for numeric robustness
6R IK, special architecture	4	Multi-hour to multi-day total	Many	Helper functions, architectural decoupling, elimination order

Appendix: Larger Slice of the 16-Solution Table

Sol	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	-148.74	-64.31	-167.76	-54.97	-88.84	44.07
2	63.90	-111.16	-126.95	47.57	-68.51	-55.40
3	-116.98	-75.52	152.12	-44.86	-100.75	105.10
4	0.00	107.46	112.46	-7.66	0.00	0.00
5	-24.19	90.31	123.04	8.55	4.18	-9.97
6	-116.98	-75.52	-27.88	-135.14	-79.25	-74.90
7	63.90	-111.16	53.05	132.43	-111.49	124.60
8	-11.16	-118.68	-38.88	138.63	-77.00	-9.11

Sol	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
9	109.73	121.72	51.12	-102.54	-3.54	1.25
10	-11.16	-118.68	141.12	41.37	-103.00	170.89
11	177.33	74.17	54.09	-150.77	-8.74	35.11
12	-148.74	-64.31	12.24	-125.03	-91.16	-135.93
13	-24.19	90.31	-56.96	171.45	175.82	170.03
14	0.00	107.46	-67.54	-172.34	180.00	-180.00
15	177.33	74.17	-125.91	-29.23	-171.26	-144.89
16	109.73	121.72	-128.88	-77.46	-176.46	-178.75

Appendix: Refined Prompt Structure for the 6R Solver

Context

Act as an expert in robot kinematics, symbolic elimination, and numerical computing. Read the three provided reference PDFs on the Raghavan–Roth method, the generalized eigenvalue strategy, and the 16-real-root benchmark case.

Task

Reconstruct the pipeline in four phases: OCR/equation extraction, Mathematica symbolic derivation, numeric Python solver, and C++ translation.

Requirements

Use FK–IK–FK validation with explicit tolerances, stress-test random robots, and verify against the published 16-real-solution benchmark.

Deliverables

Mathematica derivation script, Python and C++ solvers, and a report with exact validation residuals.