Stewart platform: Kinematics analysis and experiment results

Trung Kien Tran[1], Nghia Bao Nguyen Hoang[2], Duong Trung Ngo[2], Hoang Viet Dang[2], Duc Cuong Vu[2], Danh Huy Nguyen[2], Tung Lam Nguyen[2], Trung Kien Nguyen[3], Vu Nguyen[3]

[1] Institute of Military Technical Automation, Academy of Military Science and Technology
[2] School of Electrical and Electronic Engineering, Hanoi University of Science and Technology
[3] Academy of Military Science and Technology

Abstract

This paper presents the kinematic analysis, experimental setup, and results of a 6-DOF Stewart platform. The platform is known for high load capacity, precision, and fast response. While inverse kinematics is straightforward, forward kinematics is more challenging. We propose an inverse-kinematics-based optimization method to solve forward kinematics, implemented in C for real-time purpose. The method is also extended to a 7-DOF serial arm for validation. Experiments using EtherCAT and OMRON devices confirm the method's effectiveness for high-precision applications.
Code available at

- github.com/dc-vu/StewartPlatformForwardKinematics
- github.com/dc-vu/7DOFsInverseKinematics

State-of-the-art

- Stewart platform: 6-DOF parallel manipulator with high stiffness and accuracy; widely used in flight simulators, robotic surgery, and precision manufacturing.
- Kinematic challenges:
  - Inverse kinematics: straightforward and well-studied.
  - Forward kinematics: nonlinear, coupled, and lacks closed-form solution.
- Existing approaches:
  - Analytical methods: limited to specific configurations; may yield multiple or no real solutions.
  - Neural networks: fast inference but require large training datasets and suffer from poor generalization.
  - Optimization-based methods: formulate FK as a nonlinear least-squares problem; more adaptable to various setups.
- Current limitations: real-time implementation and experimental validation are still underexplored in existing studies.
- This work: proposes and tests optimization-based FK algorithms (Trust Region & Levenberg-Marquardt) on real hardware using EtherCAT.

Kinematics analysis

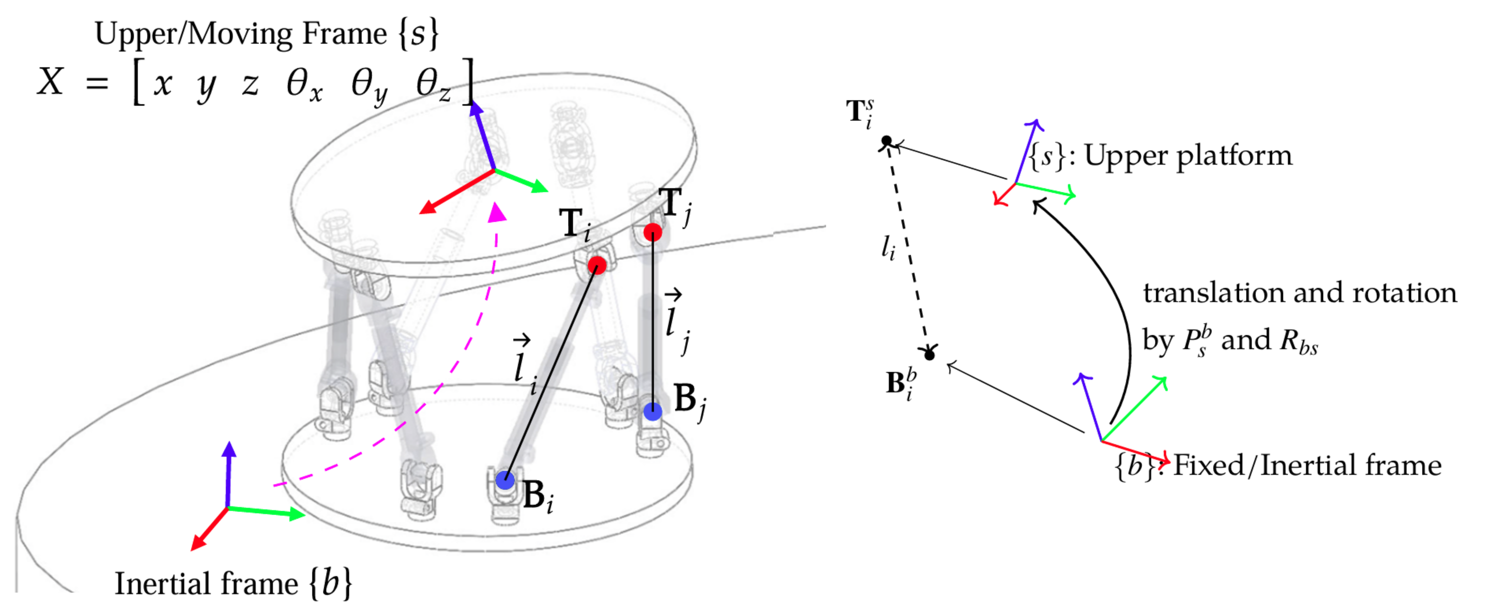

Figure 1: Kinematics analysis of Stewart platform. The red, green, and blue arrows are represented as the Ox, Oy, and Oz axis, respectively.

- A Stewart platform consists of 6 legs connecting a fixed base frame $\{b\}$ and a moving frame $\{s\}$ (Figure 1).
- Lower joints B_i and upper joints T_i are fixed in respective frames; leg lengths $l_i = \|L_i\|$ depend on pose $X = [x, y, z, \theta_x, \theta_y, \theta_z]^\top$.
- **Inverse Kinematics:** Given X, leg vectors

$$L_i = P + T_i R_{bs}(\Theta) - B_i \text{ yield } l_i = \|L_i\|$$

  $\rightarrow$ IK is well-posed and analytical.
- **Forward Kinematics:** Given $\mathcal{L}_r = [l_1, \dots, l_6]^\top$, solve

$$\mathcal{L}(P, \Theta) = \mathcal{L}_r$$

- **Optimization Formulation:**

$$g(\mathcal{L}_r) = \arg\min_{P,\Theta} \|\mathcal{L}(P, \Theta) - \mathcal{L}_r\|$$

- Trust Region and Levenberg-Marquardt methods are used for solving FK. C and Python implementations are available on GitHub.
- **Note:** Optimizers may converge to local minima. Validation of the solution is required.
- **Extension:** The same optimization principle is applied to solve inverse kinematics of a 7-DOF serial arm.

Numerical evaluation

- **Results on Stewart Platform (6-DOF):**
  - Levenberg-Marquardt is $> 50\times$ faster than Trust Region
  - Reduces iterations by $> 100\times$
  - Higher robustness and accuracy
- **Results on Serial Arm (7-DOF):**
  - $30\times$ faster in execution time
  - $50\times$ fewer iterations
  - Lower error and no failure observed
- **Hardware:** Laptop CPU Ryzen 7 5600U @ 1.7GHz

Table 1: Comparison between Trust Region and Levenberg-Marquardt methods on Inverse/Forward kinematics problems

Method	Metric	Stewart platform (6DOFs)		7DOFs serial robot arm	
		Near solution	Home position	Near solution	Home position
Trust Region	Avg. time (μs)	949.334	1157.346	3526.206	3763.735
	Avg. error	6.8×10^{-3}	15.9×10^{-3}	2.86×10^{-3}	11.6×10^{-3}
	Avg. iteration	148.26	179.5	186.32	197.67
	Failures	0%	$\approx$ 0.5%	0%	0%
LM	Avg. time (μs)	14.118	23.243	109.255	151.502
	Avg. error	88×10^{-6}	47×10^{-6}	66×10^{-6}	61×10^{-6}
	Avg. iteration	1.003	1.994	4.594	6.797
	Failures	0%	0%	0%	0%

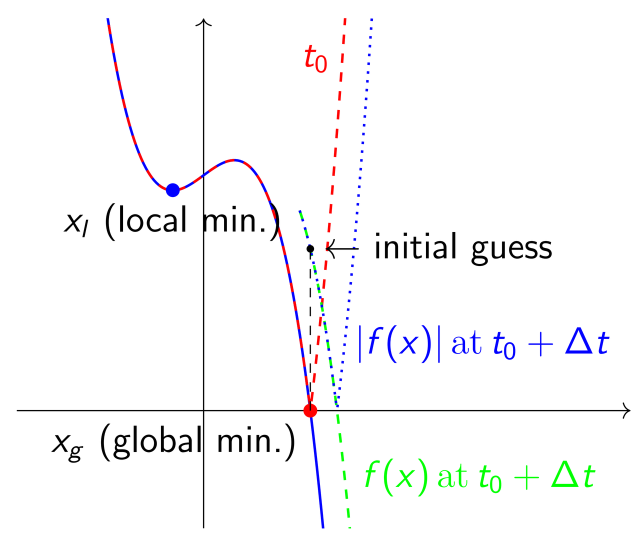

Figure 2: Local and global minimum illustration.

- The performance of the proposed algorithm could be significantly increased by using the flag -O2 when compiling by gcc. The average time of Levenberg-Marquardt can reduce to about 2-3 μs without failures.

Experiment setup and results

- **Robot:** 6-DOF Stewart platform actuated by 6 linear cylinders.
- **Actuation:**
  - Each cylinder is driven by an OMRON AC servo motor.
  - Upper joints: 3 DOFs; lower joints: 2 DOFs.
  - Prismatic joints allow full 6-DOF spatial motion.
- **Control System:**
  - Motors are controlled via OMRON servo drivers (EtherCAT slaves).
  - Master: OMRON NJ501-1300 PLC.
  - Communication with monitoring PC via Ethernet.

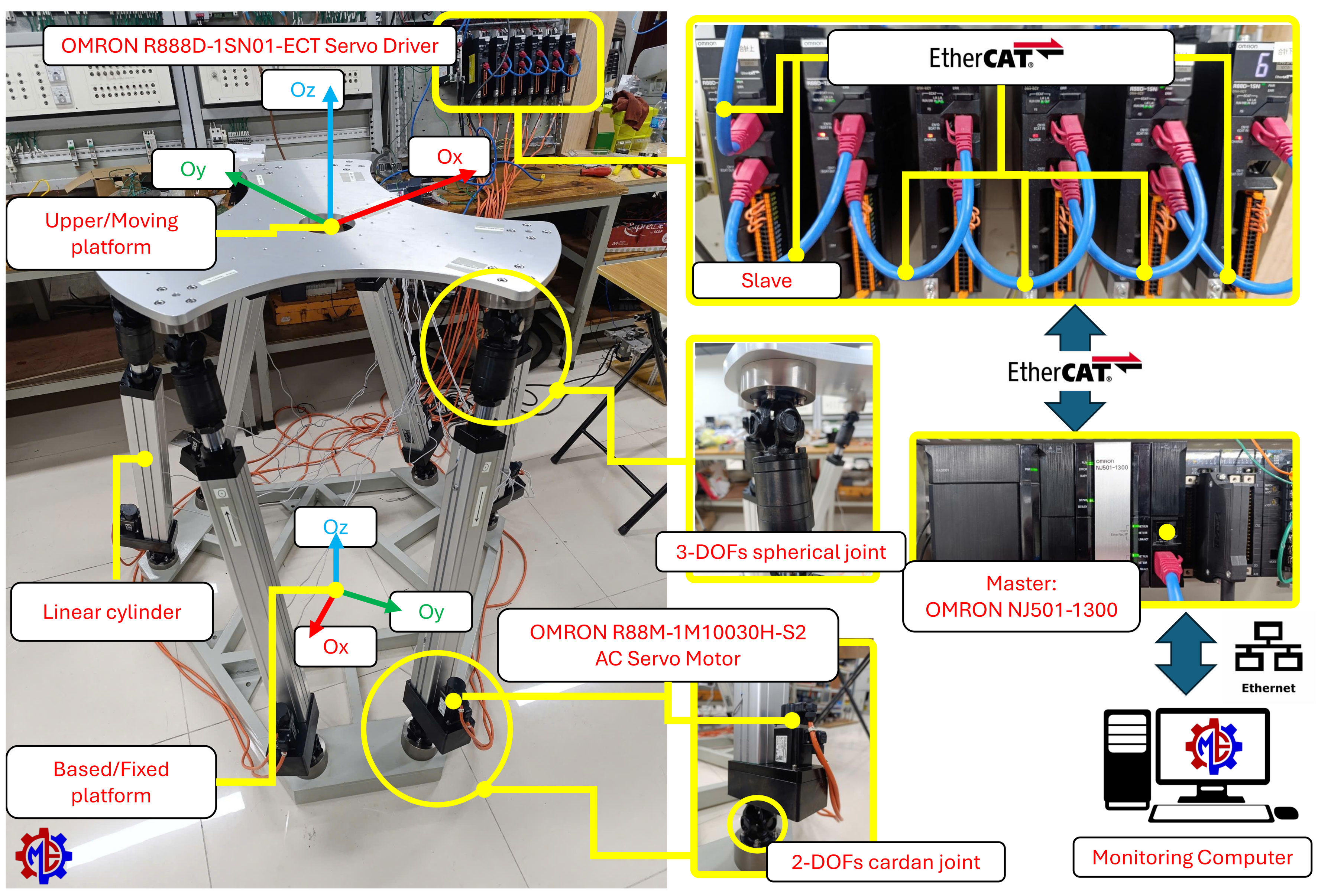

Figure 3: Experiment setup of Steawrt platform at Motion Control and Applied Robot Laboratoty (MoCAR, HUST).

- Experiment results

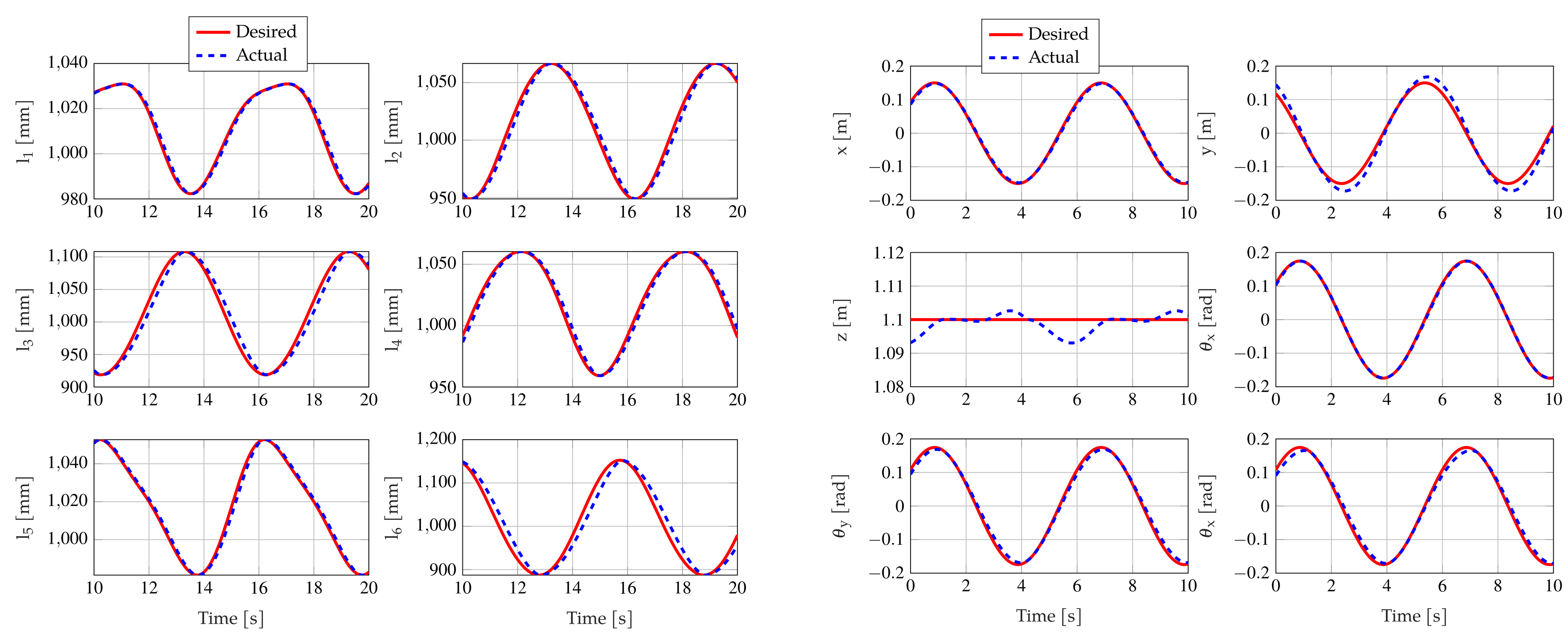

Figure 4: Experiment results of leg length tracking performance of servo controllers and forward kinematics calculated.

E-mail: vdcuong2002@gmail.com