IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Lie Group Implicit Kinematics for Redundant Parallel Manipulators: Left-Trivialized Extended Jacobians and Gradient-Based Online Redundancy Flows for Singularity Avoidance

Yifei Liu[1] and Kefei Wen[2]

***Abstract*—We present a Lie group implicit formulation for kinematically redundant parallel manipulators that yields left-trivialized extended Jacobians for the extended task variable $x = (g, \rho) \in \mathrm{SE}(3) \times \mathcal{R}$. On top of this model we design a gradient-based redundancy flow on the redundancy manifold that empirically maintains a positive manipulability margin along prescribed $\mathrm{SE}(3)$ trajectories. The framework uses right-multiplicative state updates, remains compatible with automatic differentiation, and avoids mechanism-specific analytic Jacobians; it works with either direct inverse kinematics or a numeric solver. A specialization to $\mathrm{SO}(2)^3$ provides computation-friendly first- and second-order steps. We validate the approach on two representative mechanisms: a (6+3)-degree-of-freedom (DoF) Stewart platform and a Spherical–Revolute platform. Across dense-coverage orientation trajectories and interactive gamepad commands, the extended Jacobian remained well conditioned while the redundancy planner ran at approximately 2 kHz in software-in-the-loop on a laptop-class CPU. The method integrates cleanly with existing kinematic stacks and is suitable for real-time deployment.**

***Index Terms*—Kinematics, Parallel Robots, Redundant Robots.**

I. Introduction

CLOSED-CHAIN parallel manipulators can exhibit uncontrollable internal motions near Type-II singularities, whereas open chains merely lose task-space mobility at a singularity. Recent kinematically redundant parallel manipulators promise larger singularity-free regions and better conditioning near difficult poses, by introducing three redundant DoF, often modeled as $\mathrm{SO}(2)^3$ [1], [2], [3], [4] with variants based on $\mathrm{SO}(3)$ or $\mathbb{R}^3$ [5], [6]. However, redundancy resolution that preserves feasibility along a given path has received comparatively little attention in the literature. We therefore address the pathwise problem: maintain a singularity margin along an entire prescribed $\mathrm{SE}(3)$ trajectory while remaining on the kinematic constraint manifold.

Our work builds on the classical singularity analysis of Gosselin and Angeles via partial Jacobians [7] and the coordinate-invariant formulation of end-effector kinematics on $\mathrm{SE}(3)$ by Park and Kim [8]. We model kinematics as a Lie group implicit constraint $\Phi(q, x) = \mathbf{0}$ with the redundancy-extended task variable $x = (g, \rho) \in \mathrm{SE}(3) \times \mathcal{R}$, adopt right-multiplicative state updates with left-trivialized differentials for implementation consistency. Partial Jacobians are defined with respect to right-multiplicative Lie algebra updates, implemented in a way that is compatible with automatic differentiation. We quantify the singularity margin using a smooth, differentiable singularity metric, propose a gradient-driven redundancy flow on the redundancy manifold $\mathcal{R}$.

The approach is validated on two representative mechanisms: a (6+3)-DoF Stewart platform [1] and a Spherical–Revolute (SR) platform variant [3]. Experiments show soft real-time feasibility and improved singularity margins along dense-coverage orientation trajectories and interactive commands, including tasks that exploit redundancy for additional objectives.

Manuscript received: September, 17, 2025; Revised November, 17, 2025; Accepted December, 22, 2025.

This paper was recommended for publication by Editor Pallottino, Lucia upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC)

[1]First Author and Second Author are with Department of Mechanical Engineering, Faculty of Applied Science, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada. yifei.liu@ubc.ca

[2] kwen@mech.ubc.ca

Digital Object Identifier (DOI): see top of this page.

II. Extended Kinematics and Jacobians

We follow the right-multiplicative/left-trivialized convention; Exp and Log denote the group exponential and logarithm (log is the natural logarithm). Bold uppercase denotes matrices and bold lowercase vectors; abstract group elements are italic and their matrix realizations bold. A compact symbol table is given in Appendix Table I.

Consistent with common implementations, we assume the joint space $\mathcal{J}$, the task space $\mathcal{G} = \mathrm{SE}(3)$, and the redundancy space $\mathcal{R}$ are Lie groups. Define the extended task manifold $\mathcal{X} \equiv \mathcal{G} \times \mathcal{R}$ and write $x = (g, \rho) \in \mathcal{X}$. Kinematic constraints are given by the smooth map

$$\Phi : \mathcal{J} \times \mathcal{X} \to \mathbb{R}^m, \qquad \Phi(q, x) = \mathbf{0}, \tag{1}$$

where q is the joint configuration.

Let $\xi \in \mathfrak{se}(3)$ denote the body twist. We represent the redundancy velocity by a left-trivialized algebra element $\omega \in \mathfrak{r}$. Composing the two, we have the extended task velocity algebra

$$\upsilon \equiv \begin{bmatrix} \xi \\ \omega \end{bmatrix} \in \mathfrak{se}(3) \oplus \mathfrak{r}.$$

We represent the joint velocity in a left-trivialized form: let $\nu \in \mathfrak{j}$ denote the joint velocity algebra. Linearizing Eq. (1) around a selected (q, x) yields

$$D_q^\ell \Phi \, \nu + D_x^\ell \Phi \, \upsilon = \mathbf{0}, \tag{2}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

where we define the left-trivialized partials via right-multiplicative limits

$$D_q^\ell \Phi(q,x)[\nu] \equiv \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_0 \Phi\big(q\,\mathrm{Exp}(\varepsilon\,\nu),\,x\big) \tag{3}$$

$$D_x^\ell \Phi(q,x)[\upsilon] \equiv \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_0 \Phi\big(q,\,(g\,\mathrm{Exp}(\varepsilon\,\xi),\,\rho\,\mathrm{Exp}(\varepsilon\,\omega))\big). \tag{4}$$

Fix bases for $\mathfrak{j}$, $\mathfrak{se}(3)$, and $\mathfrak{r}$. This yields the matrix form of Eq. (2); write $\mathbf{A}(q,x) \equiv D_x^\ell \Phi$ and $\mathbf{B}(q,x) \equiv D_q^\ell \Phi$. For notational brevity, from this point on we identify Lie algebra elements with their coordinate vectors via the vee map and reuse the same symbols (e.g., $\xi, \omega, \upsilon, \nu$) for their coordinate representations; accordingly, $\mathbf{A}$ and $\mathbf{B}$ act on these coordinate vectors. With this notation, following Gosselin [7], we have:

$$\mathbf{A}\,\upsilon + \mathbf{B}\,\nu = \mathbf{0}, \tag{5}$$

and the singularity criteria of Gosselin and Angeles [7] apply unchanged in this extended formulation to incorporate kinematic redundancy.

Assume dimensions are matched so that $\mathbf{A}$ and $\mathbf{B}$ are $m \times m$ matrices with $m = \dim(\mathfrak{se}(3) \oplus \mathfrak{r})$. In this extended formulation, if $\mathbf{A}$ has full rank, then a nonsingular $\mathbf{B}$ implies that both $\mathfrak{se}(3)$ and $\mathfrak{r}$ are effectively controlled; controllability of the redundant degrees of freedom is a common assumption in kinematically redundant manipulator designs. In this context, Eq. (5) allows us to locally express the joint velocity ν in terms of the extended task velocity υ:

$$\nu = -\mathbf{B}^{-1}\mathbf{A}\,\upsilon. \tag{6}$$

Automatic differentiation efficiently evaluates the $\mathbf{A}$ and $\mathbf{B}$ matrices in Eq. (5), enabling soft real-time applications without deriving explicit closed forms.

III. Redundancy Flow

Unlike conventional trajectory planning—where the $\mathrm{SE}(3)$ path is synthesized by the planner and any redundancy is handled pointwise—our setting fixes the $\mathrm{SE}(3)$ path and treats the redundancy schedule $\rho(\cdot)$ as the planning variable. For the class of kinematically redundant parallel mechanisms we study, and ignoring mechanical limits, there exists a nonsingular configuration for essentially any end-effector pose in $\mathrm{SE}(3)$. Consequently, the classical problem of generating an $\mathrm{SE}(3)$ path is not the bottleneck; the challenge is to preserve feasibility and enlarge the Type-II margin along the prescribed path by appropriately scheduling ρ.

With $\mathbf{A}$ and $\mathbf{B}$ from Eq. (5), define a singularity penalty loss function $f(g,\rho)$; minimizing f encourages avoidance of singular configurations. In analogy with the right-multiplicative limits in Eqs. (3)–(4), define the left-trivialized gradient of f with respect to ρ by

$$\langle g_\rho^L(x),\,\zeta\rangle = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_0 f\big(g,\rho\,\mathrm{Exp}(\varepsilon\zeta)\big). \tag{7}$$

If ρ follows the gradient flow in Eq. (7) from a nonsingular initial state, and the penalty f is shaped so that its local minima are nonsingular, then the trajectory remains nonsingular along the discretized sequence. A practical choice for f is a smooth log-barrier on a singularity metric; other constraints can be incorporated as needed. Examples appear in Section IV.

Let $\mathcal{U} : \mathcal{X} \times \mathbb{R}_{>0} \to \mathfrak{r}$ denote a gradient-based map that, given x and a parameter λ, e.g., a nominal step size, returns a redundancy step $s = \mathcal{U}(x;\lambda)$. A simple instance is gradient descent along $g_\rho^L(x)$. We only assume that $\mathcal{U}$ returns a stable step for singularity avoidance; concrete designs appear in Section IV.

The gradient flow above assumes the $\mathrm{SE}(3)$ component is fixed. In practice, we consider a discretized $\mathrm{SE}(3)$ trajectory with sufficiently small increments between consecutive poses. Algorithm 1 outlines the configuration planning process for such a trajectory.

Algorithm 1 Configuration Planning for Given $\mathrm{SE}(3)$ Trajectory

Require: Discretized $\mathrm{SE}(3)$ sequence $\{g_i\}_{i=0}^N$, initial redundancy ρ_0, initial joint configuration q_0
Require: Step-generator parameter λ

$i \leftarrow 0$
while $i < N$ **do**
  $\hat{x}_{i+1} \leftarrow (g_{i+1}, \rho_i)$
  $s_i \leftarrow \mathcal{U}\big(g_\rho^L(\hat{x}_{i+1});\,\lambda\big)$
  $\rho_{i+1} \leftarrow \rho_i\,\mathrm{Exp}(s_i)$
  $x_{i+1} \leftarrow (g_{i+1}, \rho_{i+1})$
  if hasDirectIK **then**
    $q_{i+1} \leftarrow \mathrm{IK}_{\mathrm{direct}}(x_{i+1})$
  else
    $\upsilon_i \leftarrow \mathrm{Log}(x_i^{-1} x_{i+1})$
    $\nu_i \leftarrow -\,\mathbf{B}(q_i,x_i)^{-1}\,\mathbf{A}(q_i,x_i)\,\upsilon_i$
    $\hat{q}_{i+1} \leftarrow q_i\,\mathrm{Exp}(\nu_i)$
    $q_{i+1} \leftarrow \mathrm{IK}_{\mathrm{numeric}}(x_{i+1}, \hat{q}_{i+1})$
  $i \leftarrow i+1$

Algorithm 1 plans the manipulator's configuration by carrying the current redundancy to the next pose and evolving along the gradient flow Eq. (7) with parameter λ. The corresponding joint configuration is then solved by inverse kinematics, abbreviated IK. If a numerical IK solver is used, $\hat{q}_{i+1} = q_i\,\mathrm{Exp}(\nu_i)$ can be used as an initial guess or for branch selection.

IV. Case Studies

Two representative kinematically redundant parallel manipulator designs are considered: the (6+3)-DoF Stewart platform by Gosselin [1] and a novel design by Wen [3]. Both admit $\mathrm{SO}(2)^3$ as their redundancy Lie group and have direct inverse kinematic solutions. By direct we mean a closed-form algebraic map $x \mapsto q(x)$, as opposed to numerical root-finding.

A. Hessian on $\mathrm{SO}(2)^3$

Let $\rho \in \mathrm{SO}(2)^3$ and let $\eta, \zeta \in \mathfrak{r} \simeq \mathbb{R}^3$ denote left-trivialized algebra directions, one per factor. For a scalar objective f :

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

$\mathcal{X} \to \mathbb{R}$ with $x = (g, \rho)$, we use the computation-friendly, left-trivialized definitions

$$\langle g_\rho^L(x),\, \zeta\rangle \equiv \left.\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\right|_0 f\big(g,\, \rho\,\mathrm{Exp}(\varepsilon\zeta)\big),$$
$$H_\rho^L(x)[\eta,\zeta] \equiv \left.\frac{\partial^2}{\partial s\,\partial t}\right|_0 f\big(g,\, \rho\,\mathrm{Exp}(t\eta)\,\mathrm{Exp}(s\zeta)\big). \quad (8)$$

These formulas are coordinate-free. For $\mathrm{SO}(2)^3$ we can exploit the product structure to obtain a simple matrix representation.

Choose the canonical basis $E_i \in \mathfrak{so}(2)$ $(i = 1, 2, 3)$ with $\mathrm{Exp}(\theta_i E_i) = R(\theta_i)$, and define a local unwrapped chart $\theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$ around ρ, with $\rho(\theta) = (R(\theta_1), R(\theta_2), R(\theta_3))$. Let $\tilde{f}(\theta) \equiv f\big(g, \rho(\theta)\big)$. Because each $\mathrm{SO}(2)$ factor is Abelian and its left Jacobian is the identity, the left-trivialized gradient and Hessian coincide with Euclidean derivatives in this chart:

$$g_\rho^L(x) \equiv \nabla_\theta\, \tilde{f}(\theta), \qquad H_\rho^L(x) \equiv \nabla^2_{\theta\theta}\, \tilde{f}(\theta).$$

Equivalently, with the canonical inner product on $\mathbb{R}^3$ and the basis $\{E_i\}$,

$$(H_\rho^L)_{ij} \;=\; H_\rho^L(x)[E_i, E_j] \;=\; \frac{\partial^2 \tilde{f}}{\partial\theta_i\,\partial\theta_j}(\theta).$$

In practice, unwrap θ at the current ρ, evaluate $\nabla_\theta \tilde{f}$ and $\nabla^2_{\theta\theta}\tilde{f}$ via automatic differentiation (AD) or closed form, and use them as the left-trivialized gradient and Hessian at ρ. This enables optional second-order steps, e.g., damped Newton or quasi-Newton, in the redundancy flow specialized to $\mathrm{SO}(2)^3$.

B. (6+3)-DoF Stewart Platform

1) Kinematic Modeling: Fig. 1 and Fig. 2 illustrate the prototype and the kinematic modeling of the mechanism proposed by Gosselin [1]. There are three redundant legs and three non-redundant legs. Vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are represented in the $x'y'z'$ frame, while all other vectors are expressed in the xyz frame.

Fig. 3 shows the geometric representation of a redundant leg in the same mechanism and was proposed by Schreiber [9]. The three redundant angles γ_i can be used to construct the redundancy product Lie group $\mathrm{SO}(2)^3$.

$$R(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \in \mathrm{SO}(2)$$
$$\rho \equiv (R(\gamma_1), R(\gamma_2), R(\gamma_3)) \in \mathrm{SO}(2)^3$$

The end-effector pose described in base frame xyz is a homogeneous transformation, which can also be denoted by $\mathrm{SE}(3)$ group element g.

$$g = {}^{xyz}\mathbf{T}_{x'y'z'} = \begin{bmatrix} {}^{xyz}\mathbf{Q}_{x'y'z'} & {}^{xyz}\boldsymbol{p}_{x'y'z'} \\ 0 & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

Let $\tilde{\boldsymbol{v}} \equiv \begin{bmatrix} \boldsymbol{v} \\ 1 \end{bmatrix} \in \mathbb{R}^4$ be the homogeneous representation of the vector $\boldsymbol{v}$. Given the end-effector pose g, the spherical joints B_i and B_j have their homogeneous coordinates in the xyz frame represented by

$$\begin{cases} \tilde{\boldsymbol{b}}_i &= g \cdot \tilde{\boldsymbol{v}}_i \in \mathbb{R}^4, \qquad \boldsymbol{b}_i \in \mathbb{R}^3 \\ \tilde{\boldsymbol{b}}_j &= g \cdot \tilde{\boldsymbol{v}}_j \in \mathbb{R}^4, \qquad \boldsymbol{b}_j \in \mathbb{R}^3 \end{cases}.$$

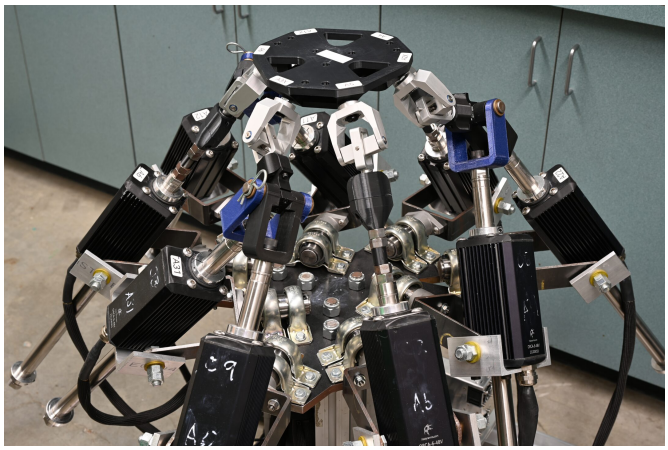

Fig. 1. A prototype of the (6+3)-DoF Stewart platform proposed by Gosselin [1].

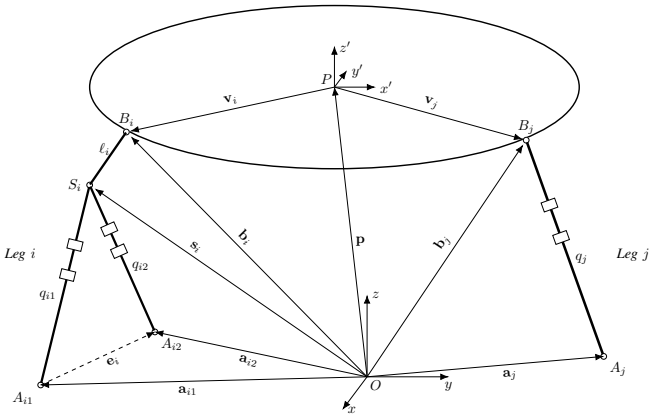

Fig. 2. Kinematic modeling of the mechanism proposed by Gosselin [1]: Only one redundant leg (leg i) and one non-redundant leg (leg j) are shown for better clarity. The joints at A_{i1}, A_{i2}, and A_j are universal joints; the joints at B_i and B_j are spherical joints; and the joint at S_i is a revolute joint.

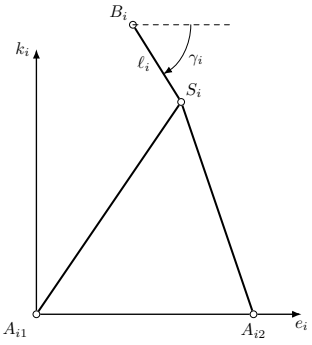

Fig. 3. Geometric representation of a redundant leg in design [1].

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Non-redundant leg j therefore has a joint coordinate given by the 2-norm of the vector from A_j to B_j:

$$q_j = \|\boldsymbol{b}_j - \boldsymbol{a}_j\|_2 \in \mathbb{R}. \tag{9}$$

For redundant leg i, we first define unit vector $\boldsymbol{e}_i \equiv \frac{\boldsymbol{a}_{i2}-\boldsymbol{a}_{i1}}{\|\boldsymbol{a}_{i2}-\boldsymbol{a}_{i1}\|_2}$. Notice that the redundant leg is a planar mechanism due to the revolute joint at S_i. With the assumption that point B_i will never align with unit vector $\boldsymbol{e}_i$, this plane is well defined by points A_{i1}, A_{i2}, and B_i. The other in-plane basis vector $\boldsymbol{k}_i$ of this plane is defined as the cross product of $\boldsymbol{e}_i$ and the vector from point A_{i1} to point B_i:

$$\boldsymbol{k}_i = \left(\boldsymbol{e}_i \times \frac{\boldsymbol{b}_i - \boldsymbol{a}_{i1}}{\|\boldsymbol{b}_i - \boldsymbol{a}_{i1}\|_2}\right) \times \boldsymbol{e}_i.$$

The point S_i is represented in the xyz frame as

$$\boldsymbol{s}_i = \boldsymbol{b}_i + \ell_i \begin{bmatrix}\boldsymbol{e}_i & \boldsymbol{k}_i\end{bmatrix} R(-\gamma_i) \begin{bmatrix}1\\0\end{bmatrix}.$$

The joint configuration of redundant leg i is given by

$$\begin{cases} q_{i1} & = \|\boldsymbol{s}_i - \boldsymbol{a}_{i1}\|_2 \in \mathbb{R} \\ q_{i2} & = \|\boldsymbol{s}_i - \boldsymbol{a}_{i2}\|_2 \in \mathbb{R} \end{cases}. \tag{10}$$

Eqs. (9) and (10) together yield the direct inverse kinematics for the mechanism, which is a function of the extended task space Lie group element $x \equiv (g, \rho)$, namely,

$$q(x) = \begin{bmatrix} q_{11}(x) \\ q_{21}(x) \\ q_{31}(x) \\ q_{12}(x) \\ q_{22}(x) \\ q_{32}(x) \\ q_1(x) \\ q_2(x) \\ q_3(x) \end{bmatrix}.$$

Since a direct inverse kinematics solution is available, the Lie group implicit function can be defined as

$$\Phi(q, x) = q - q(x) = \mathbf{0}. \tag{11}$$

Applying the left-trivialized partials to Eq. (11) yields the $\mathbf{A}$ and $\mathbf{B}$ matrices in Eq. (5). We then define $\mathbf{J} \equiv -\mathbf{B}^{-1}\mathbf{A}$ as in Eq. (6)—the inverse kinematics Jacobian. Note that this is a Lie algebra restatement of the $\mathbf{J}_e \upsilon = \mathbf{K}_e \dot{\boldsymbol{q}}$ form mentioned by Schreiber [9], where $\mathbf{K}_e$ is diagonal and nonsingular under their normalization.

2) Objective Function: We use a log-barrier on manipulability [10] together with a redundancy-limit term as the objective function.

$$f(x) \equiv -\log\det(\mathbf{J}^\top \mathbf{J}) + \mu \sum_{i=1}^{3} -\big(\log(\gamma_i - \gamma_{\min}) + \log(\gamma_{\max} - \gamma_i)\big) \tag{12}$$

The factor μ controls the weight of the redundancy-limit term. Setting $\mu = 0$ removes redundancy limits and permits angle wrapping of the revolute joints.

When the manipulator operates away from singularities, $\log\det \mathbf{J}^\top\mathbf{J}$ is concave [11]. Additionally, if operating within joint limits, f is C^2 on $\mathcal{X}$, enabling efficient gradient-based methods.

3) Step Generator: With the gradient $\boldsymbol{g} \equiv g_\rho^L(x)$ and Hessian $\mathbf{H} \equiv H_\rho^L(x)$ defined in Eqs. (7)–(8), we take a Levenberg–Marquardt style damped Newton step $s \in \mathfrak{r} \simeq \mathbb{R}^3$ by solving

$$(\mathbf{H} + \lambda I)\, s = -\boldsymbol{g}, \qquad \lambda \geq 0, \tag{13}$$

that is,

$$s = -(\mathbf{H} + \lambda I)^{-1}\, \boldsymbol{g}. \tag{14}$$

Here I is the identity with respect to the canonical inner product on $\mathfrak{r}$; for $\mathrm{SO}(2)^3$ this is the Euclidean identity $\mathbf{I}$. In practice, increase λ until $\mathbf{H} + \lambda I$ is positive definite, e.g., until Cholesky succeeds, which yields a robust step.

Eq. (14) serves as the step generator $\mathcal{U}$ mentioned in Algorithm 1.

4) Results:

a) Offline Trajectory Planning: While enlarging link lengths can expand the translational workspace, the reachable set of orientations is fundamentally bounded. To probe the orientational workspace we adopt Gosselin's tilt-torsion parameterization [1], and restate it on $\mathrm{SO}(3)$ using the group exponential. Let $\boldsymbol{e}_y \equiv [0,1,0]^\top$, $\boldsymbol{e}_z \equiv [0,0,1]^\top$, and define $R_z(\theta) \equiv \mathrm{Exp}(\theta\, \boldsymbol{e}_z)$. For azimuth b the tilt axis is $\boldsymbol{u}(b) \equiv R_z(b)\, \boldsymbol{e}_y$, and the orientation is

$$R(a,b,c) \;=\; \mathrm{Exp}\big(a\, \boldsymbol{u}(b)\big)\, R_z(c) \;\in\; \mathrm{SO}(3), \tag{15}$$

We generate smooth test trajectories by modulating (a, b, c) with sinusoids:

$$\begin{aligned} a(s) &= -\frac{A_a}{2}(\cos\big(2\pi f_a s\big) - 1), \\ b(t) &= 2\pi f_b t, \\ c(s) &= A_c \sin\big(2\pi f_c s\big), \\ t &\in [0, T] \\ s &\equiv T\left(1 - 2\left|\frac{t}{T} - \frac{1}{2}\right|\right) \end{aligned}$$

with amplitudes $A.$ defining closed intervals $[-A., A.]$. The three positive frequencies f_a, f_b, f_c are chosen mutually incommensurate, i.e., rationally independent, to avoid periodic repeats and to induce dense coverage of the attainable neighborhood around the identity. Triangular wave s is used to modulate the trajectory to have a smooth start and end at identity orientation. This yields a continuous excitation trajectory in $\mathrm{SO}(3)$.

The $\mathrm{SO}(3)$ trajectory for testing is generated with parameters $A_a = 70°$, $A_c = 20°$, $f_a = 1/\sqrt{5}$, $f_b = 1/\sqrt{7}$, $f_c = 1/\sqrt{3}$, and $T = 60$ s. Angles are specified in degrees. The end-effector is first raised in Z direction by 0.1 m in 1 s, then follows the dense-coverage orientation trajectory while holding position. Detailed dimensions of the (6+3)-DoF Stewart platform can be found in Table II in Appendix. The damped-Newton step in Eq. (14) uses $\lambda = 100$ with no weighting between translation and rotation. Time step of 1 ms is applied to discretize the trajectory. The redundancy flow algorithm is implemented in Python with JAX [12] as the automatic differentiation backend.

Fig. 4 overlays the condition number of the extended Jacobian matrix with the objective value for both methods. The

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

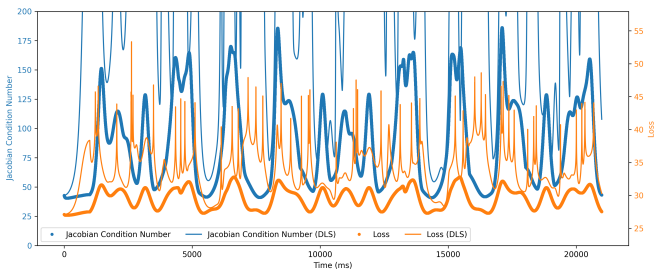

Fig. 4. Performance of the (6+3)-DoF Stewart platform with redundancy flow compared against damped least squares (DLS). The end-effector orientation follows the dense-coverage trajectory in Eq. (15). The condition-number axis is clipped at 200 for readability.

DLS baseline minimizes the joint-velocity norm at each step via the pseudo-inverse construction described by Gosselin [1], but it does not consider singularity avoidance. With redundancy flow, the condition number stays below 190 throughout the trajectory. The DLS baseline, by contrast, repeatedly enters ill-conditioned configurations, as indicated by the spikes in Fig. 4.

The step generator runs at approximately 2 kHz per planning step in the above example on an Apple M2 CPU. We therefore implemented an online gamepad control interface to allow real-time user input rather than relying on a prescribed trajectory.

b) Online Gamepad Control: Fig. 5 summarizes the signal flow. The SE(3) integrator maps u_k to a left-trivialized increment and updates g_{target}. A weighted clip $\text{Clip}_W(\cdot;\, g_{\text{act},k};\, \tau)$ limits the step relative to the current actual pose and outputs $g_{\text{act},k+1}$. Two memories store $g_{\text{target},k}$ and $g_{\text{act},k}$ (both seeded by g_0). In our open-loop setting the actual-pose memory is updated by the clip output. The redundancy-flow block computes ρ_{k+1} from $(g_{\text{act},k+1}, \rho_k)$. IK then takes $(g_{\text{act},k+1}, \rho_{k+1})$ and yields q^{ref}. Detailed definitions and equations follow.

We write $u_k \in \mathbb{R}^6$ for the normalized joystick input, Γ for fixed gains, and $\Delta t_{\text{gp}} > 0$ for the sampling period. The SE(3) increment is left-trivialized:

$$\xi_k \;=\; \Gamma\, u_k\, \Delta t_{\text{gp}}, \qquad \Delta g_k \;=\; \text{Exp}(\xi_k),$$
$$g_{\text{target},k+1} \;=\; g_{\text{target},k}\, \Delta g_k.$$

The weighted clip limits the step relative to the current actual pose; with $\delta_k \equiv \text{Log}\big(g_{\text{act},k}^{-1} g_{\text{target},k+1}\big)$ and a positive-definite weight W, define

$$\delta_k^{\text{clip}} \equiv \delta_k \cdot \min\left\{1,\; \frac{\tau}{\|W\,\delta_k\|}\right\},$$
$$g_{\text{act},k+1} \equiv g_{\text{act},k}\; \text{Exp}\big(\delta_k^{\text{clip}}\big).$$

Given the redundancy state ρ_k and actual pose $g_{\text{act},k+1}$, let the step generator return an algebra step $s_k = \mathcal{U}(g_{\text{act},k+1}, \rho_k; \lambda) \in \mathfrak{r}$, and update

$$\rho_{k+1} \;=\; \rho_k\; \text{Exp}(s_k), \qquad q_{k+1}^{\text{ref}} \;=\; \text{IK}\big(g_{\text{act},k+1},\, \rho_{k+1}\big).$$

C. Novel SR Platform

This design was first proposed by Wen [2], in which planar five-bar mechanisms serve as the three sub-legs. Delta-sub-leg [4] and XYZ-gantry [3] variants were also explored to enhance the platform's capabilities. We select the isotropic design for this case study because the shared SR mechanism can be cleanly isolated for kinematic analysis in this configuration. Fig. 6 shows a prototype of the XYZ-gantry SR platform variant.

1) Kinematic Modeling: Fig. 7 illustrates the kinematic modeling of the SR platform. This mechanism is planar: all motion lies in the platform $x'y'$-plane, and the revolute axes at B_i are parallel to the platform z'-axis.

Let constant $\tilde{\boldsymbol{v}}_i^{\,x'y'z'} \in \mathbb{R}^4$ denote the homogeneous platform-frame coordinates of point B_i, and write $g = {}^{xyz}\mathbf{T}_{x'y'z'} = \begin{bmatrix} {}^{xyz}\mathbf{Q}_{x'y'z'} & {}^{xyz}\boldsymbol{p}_{x'y'z'} \\ 0 & 1 \end{bmatrix} \in \text{SE}(3)$. The base-frame homogeneous coordinates are

$$\tilde{\boldsymbol{b}}_i \;=\; g \cdot \tilde{\boldsymbol{v}}_i^{\,x'y'z'}, \qquad \boldsymbol{b}_i \in \mathbb{R}^3.$$

where $\boldsymbol{b}_i$ is the non-homogeneous part.

Define the redundancy Lie group with per-leg phase offsets

$$\rho \equiv \big(R(\gamma_1),\, R(\gamma_2 + \tfrac{2\pi}{3}),\, R(\gamma_3 + \tfrac{4\pi}{3})\big) \in \text{SO}(2)^3.$$

With link length $\ell_i > 0$, introduce the in-plane embedding

$$\mathbf{E} \;=\; \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 2}.$$

The spherical endpoint admits the concise 3D form

$$\boldsymbol{s}_i \;=\; \boldsymbol{b}_i \;+ \mathbf{Q}\,\mathbf{E}\,\rho_i \cdot \begin{bmatrix} \ell_i \\ 0 \end{bmatrix} \in \mathbb{R}^3, \tag{16}$$

which is equivalent to the homogeneous expression

$$\tilde{\boldsymbol{s}}_i \;=\; g \cdot \begin{bmatrix} \boldsymbol{v}_i^{\,x'y'z'} + \mathbf{E}\,\rho_i \begin{bmatrix} \ell_i \\ 0 \end{bmatrix} \\ 1 \end{bmatrix}.$$

For each gantry sub-leg, fix a constant local frame $g_i = \begin{bmatrix} \mathbf{Q}_i & \boldsymbol{p}_i \\ 0 & 1 \end{bmatrix} \in \text{SE}(3)$ whose axes coincide with the three prismatic slides, forming an XYZ gantry. Let $q_i \in \mathbb{R}^3$ denote the joint vector of the three slides expressed in this local frame. Since the carriage contact point coincides with $\boldsymbol{s}_i$, the local coordinates of S_i are exactly the joint displacements. In homogeneous form this is:

$$\begin{bmatrix} \boldsymbol{q}_i \\ 1 \end{bmatrix} \;=\; g_i^{-1} \cdot \tilde{\boldsymbol{s}}_i \qquad \Longleftrightarrow \qquad \boldsymbol{q}_i \;=\; \mathbf{Q}_i^{\top}\big(\boldsymbol{s}_i - \boldsymbol{p}_i\big). \tag{17}$$

Stacking the three legs yields the full joint vector

$$q(x) \;=\; \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \end{bmatrix} \in \mathbb{R}^9,$$

which defines a direct inverse kinematics map $x \mapsto q(x)$ for this isotropic SR platform variant with XYZ-gantry.

The Lie group implicit function is again

$$\Phi(q, x) = q - q(x) = \mathbf{0}. \tag{18}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

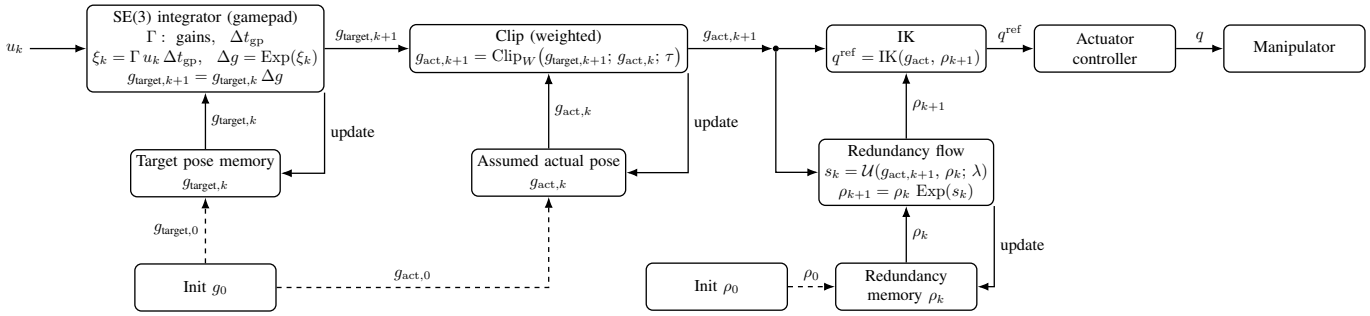

Fig. 5. Control Diagram of the Gamepad Control Interface. Loop order: read $u_k \rightarrow$ integrate $g_{\text{target}} \rightarrow$ clip to $g_{\text{act},k+1} \rightarrow$ redundancy step $\rightarrow$ update memories $\rightarrow$ IK.

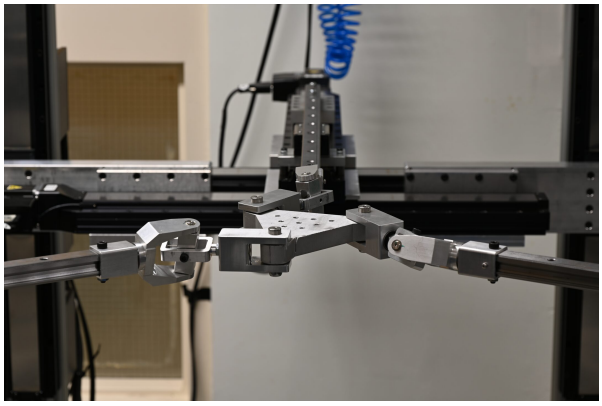

Fig. 6. A prototype of the isotropic XYZ-gantry SR platform proposed by Wen [3].

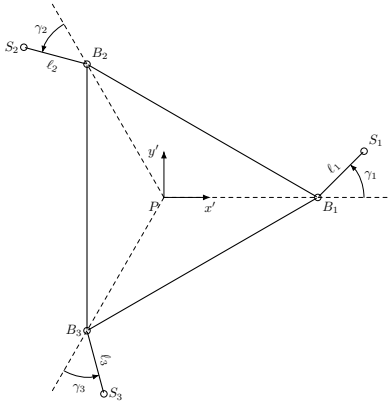

Fig. 7. Kinematic modeling of the 2D SR platform proposed by Wen [2]. The joints at B_i are revolute joints; the joints at S_i are spherical joints.

2) Objective Function and Step Generator: Since $\mathcal{R} = \mathrm{SO}(2)^3$, we reuse the (6+3)-DoF objective as in Eq. (12) and drop redundancy limits i.e. with $\mu = 0$. Because the XYZ-gantry SR variant is isotropic in $\mathrm{SE}(3)$ [3], we fix the pose g. One redundancy angle is prescribed as a continuously wrapping trajectory to stress-test singularity avoidance, while the remaining two angles are planned online by the redundancy flow.

The step generator is therefore a mapping $\mathcal{U} : \mathrm{SO}(2) \times \mathbb{R} \rightarrow \mathfrak{so}(2)^2$, with the same damped-Newton structure and parameters as in the (6+3)-DoF Stewart platform.

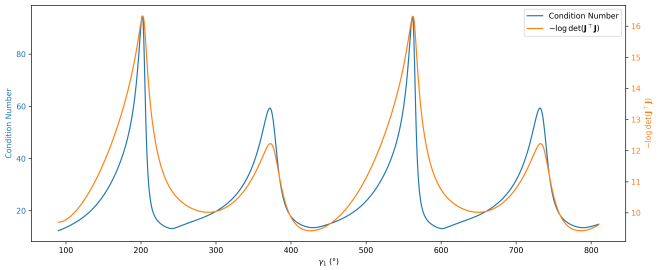

Fig. 8. Condition number of $\mathbf{J}$ and value of objective function for the SR platform with online redundancy flow, where γ_1 is prescribed as a continuously wrapping trajectory from $90°$ to $810°$, with step size $1°$.

3) Results: The revolute joints B_i are designed to be $120°$ apart on a circle of diameter 0.5. Redundant link lengths are $\ell_i = 0.2$. Fig. 8 shows the condition number and value of the objective function for the SR platform with online redundancy flow. The condition number evolution is continuous and never exceeds 100, indicating effective singularity avoidance. A video demonstration of this trajectory is included in the multimedia extension.

V. Discussion

The two case studies illustrate the versatility of the proposed framework. The Lie group implicit formulation removes the need to hand-derive analytic Jacobians for each architecture, and the redundancy flow empirically maintains a positive manipulability margin along prescribed trajectories. We therefore expect the method to generalize to other redundant parallel manipulators beyond the two studied here.

Nevertheless, singularity-avoidance performance remains mechanism-dependent: the manipulator geometry shapes the landscape of the singularity metric $f(g, \rho)$, producing broad basins in some designs and narrow ridges in others. Because the redundancy flow is local—based on first-order or damped-Newton steps—feasibility is preserved only along the kinematic branch on which the system is initialized. If that branch terminates along the prescribed $\mathrm{SE}(3)$ path, for example through self-collision, joint limits, or loss of closure, a singularity becomes unavoidable regardless of step selection.

When mechanical limits on the redundancy variables are enforced, the admissible range must be large enough to prevent branch termination during the evolution of ρ; otherwise the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

flow stalls at the boundary. Practical remedies include branch-tracking heuristics such as re-seeding, trust regions, and homotopies across redundancy charts; a systematic treatment is left for future work.

Beyond purely local updates we also tested a sliding-window, MPC-style look-ahead. Given a fixed $\mathrm{SE}(3)$ trajectory, at each index we simulate a short horizon H under several candidate redundancy-step sequences, accumulate the forecasted cost $\sum_{t=0}^{H-1} f(g_{k+t}, \rho_{k+t})$, and choose the immediate step that minimizes this prediction. The scheme improved robustness near shallow basins but increased computation and tuning effort. Integrating horizon-based selection with the present flow is a promising direction.

Finally, we note a practical application to 5-axis machining and related additive-manufacturing tasks. In these settings the tool-tip pose lies on the five-dimensional quotient manifold $\mathcal{T} \equiv \mathrm{SE}(3)/\mathrm{SO}(2)$, since spin about the tool axis is redundant. In our formulation we recast the task state as $g \in \mathcal{T}$ and augment the redundancy manifold as $\mathcal{R} \leftarrow \mathrm{SO}(2) \times \mathcal{R}$, thereby moving this spin degree of freedom from the task manifold into the redundancy variables. With this identification the redundancy-flow method applies unchanged; task-specific embeddings and objective shaping are left for future work.

VI. Conclusion

We presented a Lie group implicit formulation for redundant parallel manipulators that yields left-trivialized extended Jacobians and a gradient-based redundancy flow defined on the redundancy manifold. The method uses right-multiplicative updates, is compatible with automatic differentiation, and requires no mechanism-specific analytic Jacobians; it works with either direct inverse kinematics or a numeric solver. On a (6+3)-DoF Stewart platform and an SR platform, the approach achieved soft real-time redundancy update rates and consistently improved singularity margins along prescribed trajectories and interactive commands. The framework inserts cleanly into existing kinematic stacks, and the $\mathrm{SO}(2)^3$ specialization admits efficient Hessian steps. Future work will integrate sensing and horizon-based step selection, and study branch tracking under joint limits.

APPENDIX

TABLE I
NOTATION USED IN THE LETTER

Symbol	Meaning
G, $\mathfrak{g}$	Lie group and its Lie algebra
$g \in \mathrm{SE}(3)$	End-effector pose, abstract group element
$\rho \in \mathcal{R}$	Redundancy group element, e.g., $\mathrm{SO}(2)^3$
$q \in \mathcal{J}$	Joint configuration, group element or coordinates
$x = (g, \rho)$	Extended task variable $\mathrm{SE}(3) \times \mathcal{R}$
$\xi \in \mathfrak{se}(3)$	Body twist, left-trivialized algebra element
$\omega \in \mathfrak{r}$	Redundancy algebra element
$\nu \in \mathfrak{j}$	Joint-space algebra element
$\upsilon = [\xi; \omega]$	Extended task algebra element
Exp, Log	Group exponential and logarithm
$\mathrm{Log}_y^L(z)$, $\mathrm{Retr}_y^L(\xi)$	Left-trivialized local log $\mathrm{Log}(y^{-1}z)$ and retraction $y\,\mathrm{Exp}(\xi)$
$\widehat{\cdot}$, $(\cdot)^\vee$	Hat/vee identification, standard robotics convention
$\mathbf{A}$, $\mathbf{B}$	Left-trivialized partials in Eq. (5)
$\mathbf{J} = -\mathbf{B}^{-1}\mathbf{A}$	IK Jacobian in Eq. (6)
$\mathrm{Clip}_W(\cdot;\cdot;\tau)$	Weighted clip operator, online control

TABLE II
DIMENSIONS OF THE (6+3)-DOF STEWART PLATFORM

Vector	x [m]	y [m]	z [m]
$\boldsymbol{a}_1$	-0.270	0.200	0.000
$\boldsymbol{a}_2$	0.308	0.134	0.000
$\boldsymbol{a}_3$	-0.038	-0.334	0.000
$\boldsymbol{a}_{11}$	-0.270	0.000	0.000
$\boldsymbol{a}_{21}$	0.135	0.234	0.000
$\boldsymbol{a}_{31}$	0.135	-0.234	0.000
$\boldsymbol{a}_{12}$	-0.270	-0.275	0.000
$\boldsymbol{a}_{22}$	-0.103	0.371	0.000
$\boldsymbol{a}_{32}$	0.373	-0.096	0.000
$\boldsymbol{v}_1$	-0.111	0.072	0.000
$\boldsymbol{v}_2$	0.117	0.060	0.000
$\boldsymbol{v}_3$	-0.007	-0.132	0.000
$\boldsymbol{v}_4$	-0.111	0.072	0.000
$\boldsymbol{v}_5$	-0.007	0.132	0.000
$\boldsymbol{v}_6$	0.117	0.060	0.000

References

[1] C. Gosselin and L.-T. Schreiber, "Kinematically redundant spatial parallel mechanisms for singularity avoidance and large orientational workspace," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 286–300, Apr. 2016.

[2] K. Wen, T. S. Nguyen, D. Harton, T. Laliberté, and C. Gosselin, "A backdrivable kinematically redundant (6+3)-degree-of-freedom hybrid parallel robot for intuitive sensorless physical human–robot interaction," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1222–1238, Aug. 2021.

[3] K. Wen, Q. Ma, Y. Zhang, Y. Liu, and J. H. Sun, "Kinematic analysis of an isotropic kinematically redundant (6+3)-DOF hybrid parallel robot," in *Proceedings of the 2025 CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, E. Lanteigne and S. Nokleby, Eds. Cham: Springer Nature Switzerland, 2025, pp. 225–234.

[4] Q. Ma, Y. Quan, and K. Wen, "Kinematic analysis and workspace enhancement of a novel (6+3)-DOF kinematically redundant parallel robot," in *Proceedings of the 2025 CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, E. Lanteigne and S. Nokleby, Eds. Cham: Springer Nature Switzerland, 2025, pp. 212–224.

[5] G. Aruquipa, P. Lambert, and C. Gosselin, "Kinematic analysis and design of a novel 9-DOF parallel robot with grasping capabilities," in *Proceedings of the 2025 CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, E. Lanteigne and S. Nokleby, Eds. Cham: Springer Nature Switzerland, 2025, pp. 117–127.

[6] X. Liang, X. Zeng, G. Li, W. Chen, T. Su, and G. He, "Design, analysis, and optimization of a kinematically redundant parallel robot," *Actuators*, vol. 12, no. 3, p. 120, Mar. 2023.

[7] C. Gosselin and J. Angeles, "Singularity analysis of closed-loop kinematic chains," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 281–290, June 1990.

[8] F. C. Park and J. W. Kim, "Singularity analysis of closed kinematic chains," *Journal of Mechanical Design*, vol. 121, no. 1, pp. 32–38, Mar. 1999.

[9] L.-T. Schreiber and C. Gosselin, "Exploiting the kinematic redundancy of a (6+3) degrees-of-freedom parallel mechanism," *Journal of Mechanisms and Robotics*, vol. 11, no. 2, p. 021005, Feb. 2019.

[10] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, June 1985.

[11] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[12] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/jax-ml/jax