2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Surface Following using Direct Adaptive Admittance Control

Iñigo Iturrate, Emil Lykke Diget and Christoffer Sloth

***Abstract*— Many robotic tasks, such as polishing or grinding, involve maintaining contact with and applying a force against the environment while following a given trajectory. In this paper, we present an adaptive admittance controller that aligns its control parameters online to be in the direction of an estimate of the surface normal vector. This essentially allows a robot to follow an unknown surface, as is the case in uncalibrated setups or quick changeover production. We present and compare three different surface normal estimation algorithms: the integral adaptive law and two Riemannian manifold based algorithms.**

Our experimental results show that the adaptive controller using the simple Riemannian gradient descent yields the lowest tracking error of the three. It has $73\,\%$ decrease in positional error and $43\,\%$ decrease in angular error compared with the controller with the integral adaptive law, and overall is effective at aligning the robot tool online against surface moving in an a priori unknown pattern.

I. Introduction

As the demand for low-volume production increases, there is a need for robots that are can be rapidly reprogrammed. One approach to rapid robot programming is kineasthetic teaching, which is a no-code programming approach that lets non-programmers setup robots by simply grabbing them and moving them to a position or through a path. Many rapid reprogramming approaches, including kineasthetic teaching, do not rely on a-priori knowledge about the geometry of objects to be manipulated. Therefore, there is a need for robot control that allows interaction with objects having unknown geometry. This type of controllers often rely on online contact point and surface normal estimation.

Surface normal estimation has been studied in various contexts within robotics. In [1], contact point and surface normal estimation was used in the context of using uncalibrated tools; that work was generalized in [2] to consider a soft finger contact model that includes torsional friction. Both methods rely on constant parameter estimation using an integral adaptive law [3]. If sensors that provide local surface geometry of an objects in terms of a contact patch are available, then object pose estimation is possible based on the contact information [4]. In applications such as a gluing task, as studied in [5], [6], the curvature of the environment can be unknown and it has to be estimated using e.g., a surface normal estimator using tactile measurements.

Much work has been conducted on adaptive admittance control in different contexts. In [7], an adaptive admittance control was proposed for human–robot interaction. A PD-controller and indirect adaptive update strategy was proposed to improve the force tracking accuracy. Similarly, [8] proposed an online indirect adaptive update strategy to achieve a good force tracking in steady state. In [9], an adaptive neural controller was proposed to guarantee trajectory tracking. Safety plays an important role in robot-environment interaction; thus, [10] used exponential control barrier functions to ensure the safety of an admittance control.

Iñigo Iturrate, Emil Lykke Diget and Christoffer Sloth are with the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Denmark {inju,eld,chsl}@mmmi.sdu.dk.

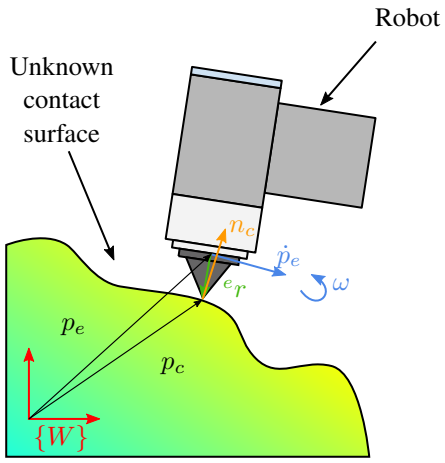

Fig. 1. The robot in contact with the environment.

In this paper, direct adaptive control, whereby estimated parameters are used directly in the controller, will be used for the proposed adaptive admittance control. For this class of adaptive controllers, the convergence properties of the utilized parameter estimator are pivotal for the overall system performance. Numerous parameter estimators have been proposed with different properties [3]. Recently, dynamic regressor extension and mixing (DREM) was proposed in [11] to improve and decouple the convergence of parameter estimates. Also, parameter estimation with geometric constraints has been considered to improve the parameter estimation algorithms [12].

In this paper, an adaptive admittance controller is proposed that allows robots to follow the surface of an object with unknown geometry. The controller uses measurements of motion and force to estimate the surface normal of the object; this is used for adapting the controller parameters and reference trajectories. Fast convergence of the surface normal estimate is pivotal for obtaining good surface following. Therefore, an estimation algorithm based on Riemannian gradient descent is proposed, and it shown to be superior to the standard method using the integral adaptive law.

The paper is organized as follows. An overview of the system is provided in Section II. Section III provides

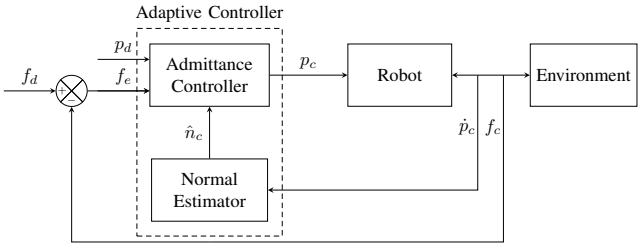

Fig. 2. Block diagram of the positional part of the control scheme for online alignment to unknown surfaces. The angular part is similar.

background on parameter estimation. The proposed adaptive control is presented in Section IV. An experiment where the proposed method for robotic surface treatment is applied to a moving surface is presented in Section V. Lastly, conclusions are presented in Section VI.

II. System Overview

Here, we will present an overview of the paper. On Fig. 1, a diagram of the interaction between the robot and the environment can be seen. The tool with the tip described by ${}^{e}r$ is in contact with a stiff environment. The robot is controlled to move around on the surface with a velocity $\dot{p}_c$ and a rotational velocity ω. The normal direction of the surface is n_c and its estimate is $\hat{n}_c$.

An overview of the control scheme can be seen on Fig. 2. The normal estimator estimates the surface normal based on measurements of the velocity, and is described in Section III. Then, the adaptive admittance controller, described in Section IV, uses the estimated normal vector together with a desired force f_d and a desired pose, described by a position p_d and a rotation R_d, to compute the contact pose p_c, R_c.

III. Surface Normal Estimation

In this section, we present three methods for estimating the surface normal. A diagram of the system can be seen on Fig. 1 and forms the basis of the system model described here.

Assume that the tool tip of a robot is in contact with the environment at all times, then the following condition is true:

$$n_c^\intercal \dot{p}_c = 0, \tag{1}$$

where $n_c \in \mathbb{R}^3$ is the surface normal vector, and $\dot{p}_c$ is the velocity of the contact point, which is given by

$$\dot{p}_c = \dot{p}_e + S(\omega) R_e {}^{e}r, \tag{2}$$

where $\dot{p}_e \in \mathbb{R}^3$ is the end-effector velocity, $R_e \in \mathbb{R}^{3\times3}$ is the orthonormal rotation matrix from base frame to end-effector frame, ${}^{e}r$ is the contact-point in the end-effector frame, and S is the skew-symmetric operator defined as

$$S(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3}$$

The first method uses the integral adaptive law to estimate the surface normal, and is a summary of the methodology of [13], [14]. Then, the two remaining methods are based on optimisation on a Riemannian manifold. The normal vector is a unit vector, which is a point on the 2-sphere embedded in $\mathbb{R}^3$. The first is called Riemannian gradient descent, and the next is Riemannian gradient descent with regressor extension.

A. Integral Adaptive Law with Sphere Projection

In [13], [14] the integral adaptive law [3] was used to estimate the parameter n_c as follows:

$$\dot{\hat{n}}_c(t) = -\gamma P(\hat{n}_c(t)) L(t) \hat{n}_c(t) \tag{4}$$

$$\dot{L}(t) = -\beta L(t) + \frac{1}{1 + \|\dot{p}_c(t)\|^2} \dot{p}_c(t) \dot{p}_c^\intercal(t), \quad L(0) = 0, \tag{5}$$

where $P(v) = I_3 - vv^\intercal$ is a projection matrix that ensures that v is a unit vector, I_3 is the 3×3 identity matrix, and $\gamma, \beta > 0$ are controller gains. While this formulation has the projection matrix P to keep the normal vector at unity length, i.e., as a point on a 2-sphere embedded in $\mathbb{R}^3$. Alternatively to the projection matrix, the normal vector can be considered a point on the Riemannian manifold of 2-spheres, and a estimator can be formulated using Riemannian operators.

B. Riemannian Gradient Descent

To avoid the necessity of projecting the normal vector down to the sphere explicitly a, perhaps more elegant, way is to consider it as a point on the Riemannian manifold of 2-spheres as defined by

$$\mathcal{S}^2 \triangleq \{v \in \mathbb{R}^3 : \|v\| = 1\}. \tag{6}$$

A smooth d-dimensional manifold $\mathcal{M}$ is a topological space that locally behaves as the Euclidean space $\mathbb{R}^d$. There exists a tangent space $\mathcal{T}_p\mathcal{M}$ for each point $p \in \mathcal{M}$ that locally linearises the manifold. A Riemannian manifold has a positive definite metric tensor, also called the inner product, on each tangent space $\mathcal{T}_p\mathcal{M}$ denoted as $\langle \cdot, \cdot \rangle_p$ also known as the Riemannian metric [15].

It is possible to map between the manifold and its tangent space. The exponential map $\mathrm{Exp}_p : \mathcal{T}_p\mathcal{M} \to \mathcal{M}$ maps a point u in the tangent space to a point on the manifold. The inverse map is called the logarithmic map $\mathrm{Log}_p : \mathcal{M} \to \mathcal{T}_p\mathcal{M}$ [15].

In Euclidean space, the exponential map is $\mathrm{Exp}_p(v) = q = p + v$, and the logarithm map is $\mathrm{Log}_p(q) = v = q - p$. The definitions of these maps can easily be looked up for the applied manifold, e.g., in a software library such as `manopt` [16].

Then, consider the condition of the normal vector (1), and define the Euclidean quadratic cost function as

$$\bar{f}(\hat{n}_c) = \frac{1}{2} \|\hat{n}_c^\intercal \dot{p}_c\|^2, \tag{7}$$

which can be minimised by a gradient-based parameter estimator [17], i.e., taking a step in the negative direction of the gradient with respect to $\hat{n}_c$. The Euclidean gradient is then:

$$\mathrm{grad}\, \bar{f}(\hat{n}_c) = (\hat{n}_c^\intercal \dot{p}_c) \dot{p}_c^\intercal. \tag{8}$$

Though, when optimising on a Riemannian manifold, the gradient is defined as:

Definition 1 (Definition 3.58 in [15])**.** Let $f : \mathcal{M} \to \mathbb{R}$ be smooth on a Riemannian manifold $\mathcal{M}$. The Riemannian gradient of f is the vector field $\operatorname{grad} f$ on $\mathcal{M}$ uniquely defined by

$$\mathrm{D}f(x)[v] = \langle v, \operatorname{grad} f(x) \rangle_x \quad \forall (x, v) \in \mathcal{T}_x\mathcal{M}, \tag{9}$$

where $\mathrm{D}f(x)$ is the directional derivative of $f(x)$ along v.

Inspired by the gradient descent in Euclidean space, the Riemannian gradient descent (RGD) for optimising on the manifold is considered [18]:

$$\hat{n}_c(t) = \operatorname{Exp}_{\hat{n}_c(t-T_s)} \Big(-\gamma T_s \operatorname{grad} f(\hat{n}_c) \Big), \tag{10}$$

discretized using an Euler step with time-step T_s, $\gamma > 0$ is a tuning parameter, and $\operatorname{grad} f(\hat{n_c}) : \mathcal{M} \to \mathcal{T}_{\hat{\theta}(t)}\mathcal{M}$ is the Riemannian gradient, which, in the case of a spherical Riemannian manifold is defined as [15]

$$\operatorname{grad} f(\hat{n}_c) = \operatorname{grad} \bar{f}(\hat{n}_c) - \left(\hat{n}_c^\intercal \operatorname{grad} \bar{f}(\hat{n}_c)\right) \hat{n}_c. \tag{11}$$

An implementation of the general Riemannian gradient descent was implemented in the C++-library `sdu_estimators`[1].

C. Riemannian Gradient Descent with Regressor Extension

To improve the convergence properties of gradient estimators, it was proposed in [19] to apply a $\mathcal{L}_\infty$ operator $\mathcal{H}$ filter the linear regression equation as follows. Applied to the problem of surface normal estimation, it is

$$\mathcal{P}(t) = \mathcal{H}[\dot{p}(t)] \in \mathbb{R}^{3\times 3}, \tag{12}$$

resulting in the filtered equation

$$\mathcal{P}^\intercal(t) n_c(t) + \varepsilon(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{13}$$

where ε is a filtering error that is related to the derivative of the parameter n_c. The rest of the procedure is in the same fashion as in with standard Riemannian gradient descent, i.e., by defining the quadratic cost function

$$\bar{f}(n_c) = \frac{1}{2} \left\| \mathcal{P}^\intercal n_c \right\|^2, \tag{14}$$

with the Euclidean gradient:

$$\operatorname{grad} \bar{f}(n_c) = \mathcal{P}(\mathcal{P}^\intercal n_c), \tag{15}$$

from which the Riemannian gradient and the subsequent parameter estimate can be estimated in the same fashion as in the prior section.

[1] `https://github.com/SDU-Robotics/sdu_estimators`

Algorithm 1 Align z-axis orientation to surface normal

Require: normal estimate n_c, initial rotation matrix R_{ref}
Ensure: aligned rotation matrix R_{aligned}

procedure ALIGNZTOVECTOR(n_c, R_{ref})
  $x_{\text{old}} \leftarrow R_{\text{ref}}[:, 1]$ ▷ Select x-axis of rotation matrix
  $y_{\text{old}} \leftarrow R_{\text{ref}}[:, 2]$ ▷ Select y-axis of rotation matrix
  $z_{\text{new}} \leftarrow n_c$ ▷ Align z-axis with surface normal
  $x_{\text{new}} \leftarrow \text{normalize}\,(y_{\text{old}} \times z_{\text{new}})$ ▷ Calculate shortest rotation
  $y_{\text{new}} \leftarrow \text{normalize}\,(z_{\text{new}} \times x_{\text{new}})$
  $R_{\text{aligned}} \leftarrow \begin{bmatrix} x_{\text{new}} & y_{\text{new}} & z_{\text{new}} \end{bmatrix}$
  return R_{aligned}

IV. Control

This section will describe the adaptive control scheme for online alignment to unknown surfaces. A block diagram of the system can be seen in Fig. 2. The basis for the system is an adaptive controller consisting of an admittance controller with a feedforward force and a normal estimator. The adaptation law automatically handles alignment of the robot tool tip to a surface of unknown curvature that it is assumed to be in contact with, while simultaneously ensuring the gain matrices of the admittance controller are rotated accordingly.

The admittance controller in Fig. 2 is implemented after [20] as follows:

$$M_p \ddot{x}_e + D_p \dot{x}_e + K_p x_e = f_e, \tag{16}$$

where $M_p, D_p \in \mathbb{R}^{3\times 3}$ are positive definite matrices describing the virtual mass and damping of the robot end-effector, and $K_p \in \mathbb{R}^{3\times 3}$ is its stiffness, which is positive semi-definite. The term $x_e \in \mathbb{R}^3$ is defined as $x_e = x_c - x_d$, where x_c, the compliant frame, corresponds to the position of the robot end-effector, and x_d, the desired frame, is a controller input. The force error, $f_e \in \mathbb{R}^3$ is defined as $f_e = f_d - f_c$, where f_c is the measured force at the compliant frame, and f_d is a desired feedforward force that the robot should apply against the environment. An analogous version of Eq. 16 is implemented using quaternions for control of the orientation, see [21].

In order to both align the robot end-effector to the target unknown surface, as well as to ensure that the target feedforward force, f_d, is applied in the correct direction, the target robot orientation is updated online according to Algorithm 1, and the controller gain parameters, M_p, D_p, K_p, are modulated online according to Algorithm 2. In both cases, it is assumed, without loss of generality, that the robot tool contacts the environment in the direction of its z-axis.

V. Experiment

In this section, the presented adaptive control algorithm is tested in an experiment and compared with the prior method using the integral adaptive law [13]. The experimental setup can be seen in Fig. 3, where one robot is holding a plate such that the true normal vector is known, and the other robot is running the adaptive control algorithm to follow the

Algorithm 2 Align gain matrix to surface normal

Require: normal estimate n_c, gain value in unconstrained subspace k_{free}, gain value in constrained subspace k_{contact}
Ensure: aligned gain matrix K_{aligned}

procedure ALIGNGAINMATRIXTOVECTOR(n_c, k_{free}, k_{contact})
  $o_{n_c} \leftarrow \text{null}(n_c)$ ▷ Compute null-space of n_c
  $K_{\text{aligned}} \leftarrow \text{zeros}(3, 3)$
  for $i = 1, 2$ **do**
    $p \leftarrow o_{n_c} \cdot o_{n_c}^T$
    $K_{\text{aligned}} \leftarrow K_{\text{aligned}} + p \cdot k_{\text{free}}$ ▷ Assign k_{free} to null-space directions
  $p \leftarrow o_{n_c}^T \cdot o_{n_c}$
  $K_{\text{aligned}} \leftarrow K_{\text{aligned}} + p \cdot k_{\text{contact}}$ ▷ Assign k_{contact} to constrained direction
  return K_{aligned}

Fig. 3. The experimental setup. The robot on the right is holding and rotating the plane, and the robot on the left is moving circularly while controlling its z-axis towards the estimated surface normal.

normal vector. With this setup we can easily compare the true surface normal vector with the estimated one. Note that this is done purely for the sake of the experiment, and the proposed method is valid for an arbitrary unknown curved surface.

We compare the same adaptive control algorithm with different surface normal estimation algorithms. We compare the integral adaptive law (IAL) (4), the Riemannian gradient descent with a quadratic cost function (RGD) (10), and the Riemannian gradient descent with regressor extension (RGD-RE) (15). Here we use the Kreisselmeier regressor extension [22], hence the abbreviation. The KRE is simple to implement, as it only has one coefficient a, and it preserves the system excitaion from ϕ to Δ [23]. The gains and coefficients of the three surface normal estimation algorithms can be found in Table I. Note that the values of the parameters are not directly comparable for different estimators.

The robot is instructed to move in a circular pattern on the surface of the plate for 60 seconds, as can be seen on Fig. 3 and, for illustrative purposes on Fig. 4. The robot holding the plate is following a sinusoidal signal with an amplitude of $\pi/12\,\text{rad}$ and a frequency of $1/(4\pi)\,\text{Hz}$. The controlled robot follows a circle with a radius of $50\,\text{mm}$ with a frequency of $0.05\,\text{Hz}$. The end-effector velocity is used to estimate the normal vector using the mentioned algorithms and the robot is controlled as described in Section IV with the parameters found in Table II.

TABLE I
THE PARAMETERS OF THE ESTIMATION ALGORITHMS.

	Parameter	Value
IAL	γ	1
	β	0.8
RGD	γ	250
RGD-RE	γ	1 000 000
	a	5

TABLE II
PARAMETERS OF THE CONTROLLER DURING THE EXPERIMENT.

Position		Orientation	
Parameter	Value	Parameter	Value
$k_{\text{p,free}}$	270	$k_{\text{o,free}}$	50
$k_{\text{p,contact}}$	0	$k_{\text{o,contact}}$	50
$d_{\text{p,free}}$	500	$d_{\text{o,free}}$	150
$d_{\text{p,contact}}$	225	$d_{\text{o,contact}}$	100
$m_{\text{p,free}}$	2.5	$m_{\text{o,free}}$	0.25
$m_{\text{p,contact}}$	2.5	$m_{\text{o,contact}}$	0.25

The results of an experiment can be seen on Fig. 6 and Fig. 7, where the angular and the positional error, respectively, for the first 20 seconds can be seen.

The angular error is calculated as the Riemannian distance between the estimated normal vector and the true one and the unit is in degrees. The position error is the Euclidean distance of the actual and desired position projected into the plane it moves on, see Fig. 5, while the position in the world frame can be seen on Fig. 4. Both for angular and positional error, the standard RGD has the smallest error, where the numbers can be seen in Table III.

The RGD has a positional error of $(2.53 \pm 2.97)\,\text{mm}$ which is lower than both IAL and RGD-RE. Curiosly enough, RGD-RE has the biggest positional error $(14.20 \pm 9.16)\,\text{mm}$ compared with IAL with $(9.42 \pm 6.00)\,\text{mm}$. RGD and RGD-RE both have low angular error with $(3.11 \pm 1.56)^\circ$

TABLE III
THE MEAN AND THE STANDARD DEVIATION OF THE ERROR FOR THE POSITIONAL AND THE ANGULAR TRACKING OF THE CONTROLLER WITH A GIVEN SURFACE NORMAL ESTIMATOR. IT CAN BE SEEN, THAT THE RIEMANNIAN GRADIENT DESCENT WITHOUT REGRESSOR EXTENSION PROVIDES THE TRACKING WITH THE SMALLEST ERROR.

Estimator	Pos. err. $(\mu \pm \sigma)$ [mm]	Angl. err. $(\mu \pm \sigma)$ [°]
IAL	9.42 ± 6.00	5.47 ± 2.60
RGD	**2.53 ± 2.97**	**3.11 ± 1.56**
RGD-RE	14.20 ± 9.16	3.43 ± 1.77

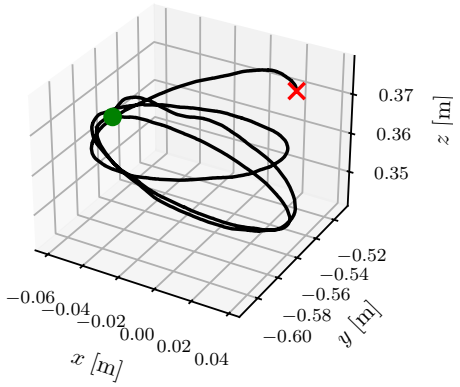

Fig. 4. The position trajectory of the first 10 s of the experiment using the Riemannian gradient descent estimator. The position of the tool tip in world frame. The green dot is the initial position, and the red cross is the position after 10 s.

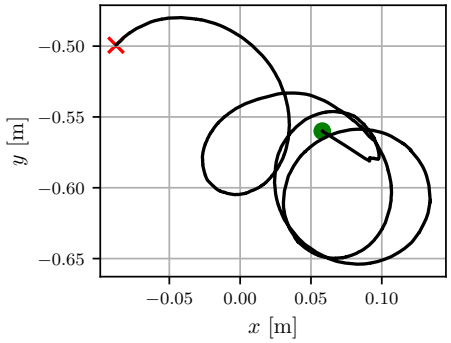

Fig. 5. The position trajectory of the first 10 s of the experiment using the Riemannian gradient descent estimator. The position of the tool tip projected into the plane. The green dot is the initial position, and the red cross is the position after 10 s.

and $(3.43 \pm 1.77)°$, respectively, compared with IAL with $(5.47 \pm 2.60)°$.

It can be seen that by using the RGD-based estimator instead of the IAL-based estimator, performance is increased in both the positional as well as the angular error. For this experiment, we see a 73 % decrease in position error, and a 43 % decrease in angular error. This is to be expected, as the RGD-based estimator computes the parameter update in a lower-dimensional manifold, which should lead to faster convergence. However, the RGD-based estimator with regressor extension has a large positional error compared with the other two. This requires further research.

VI. Conclusion

We presented an adaptive admittance controller that is capable of aligning to the estimated normal direction of an unknown surface. We presented three different surface normal estimation algorithms; the integral adaptive law and two Riemannian manifold based algorithms. The first is Riemannian gradient descent with a quadratic cost function. The second is the same but with regressor extension of the

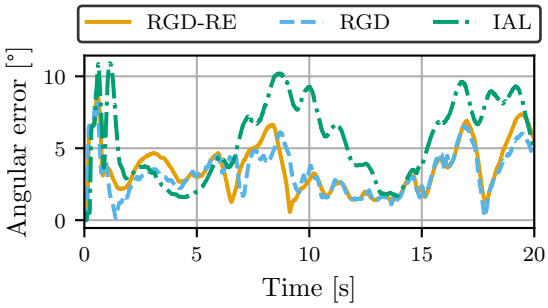

Fig. 6. Angular error. The IAL-based controller has clearly the highest angular error.

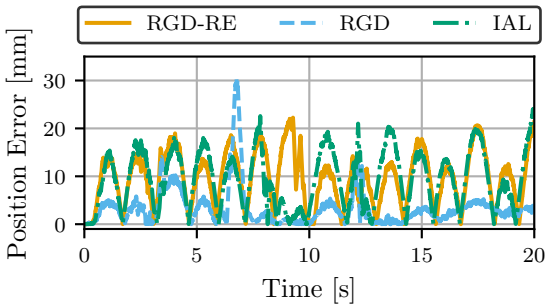

Fig. 7. Position error. The RGD-based controller has the smallest positional error.

regression equation.

From an experiment, we found that the adaptive controller using the standard Riemannian gradient descent yields the best performance of the three. It has 73 % decrease in positional error and 43 % decrease in angular error compared with the controller with the integral adaptive law. We also find that the controller with Riemannian gradient descent with regressor extension yields the highest positional error, probably due to filtering artefacts. This should be investigated as part of future research.

Acknowledgement

The work undertaken in this paper was supported by the SDU I4.0Lab at the University of Southern Denmark.

References

[1] Y. Karayiannidis, C. Smith, F. E. Vina, and D. Kragic, "Online contact point estimation for uncalibrated tool use," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2488–2494, IEEE, 2014.

[2] C. Sloth and I. Iturrate, "Simultaneous contact point and surface normal estimation during soft finger contact," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 19–25, IEEE, 2021.

[3] P. A. Ioannou and J. Sun, *Robust adaptive control*. Courier Corporation, 2012.

[4] N. Kuppuswamy, A. Castro, C. Phillips-Grafflin, A. Alspach, and R. Tedrake, "Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1811–1818, 2019.

[5] I. n. Iturrate, A. Kramberger, and C. Sloth, "Quick setup of force-controlled industrial gluing tasks using learning from demonstration," *Frontiers in Robotics and AI*, vol. 8, 11 2021.

[6] C. Sloth, A. Kramberger, E. L. Diget, and I. Iturrate, "Towards contact point and surface normal estimation for control of flexible tool," in *2022 American Control Conference (ACC)*, IEEE, 6 2022.

[7] I. Ranatunga, F. L. Lewis, D. O. Popa, and S. M. Tousif, "Adaptive admittance control for human–robot interaction using model reference design and adaptive inverse filtering," *IEEE transactions on control systems technology*, vol. 25, no. 1, pp. 278–285, 2016.

[8] C. Liu and Z. Li, "Force tracking smooth adaptive admittance control in unknown environment," *Robotica*, vol. 41, no. 7, pp. 1991–2011, 2023.

[9] C. Yang, G. Peng, Y. Li, R. Cui, L. Cheng, and Z. Li, "Neural networks enhanced adaptive admittance control of optimized robot–environment interaction," *IEEE transactions on cybernetics*, vol. 49, no. 7, pp. 2568–2579, 2018.

[10] Y. Sun, M. Van, S. McIlvanna, N. N. Minh, S. McLoone, and D. Ceglarek, "Adaptive admittance control for safety-critical physical human robot collaboration," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1313–1318, 2023. 22nd IFAC World Congress.

[11] S. Aranovskiy, A. Bobtsov, R. Ortega, and A. Pyrkin, "Performance enhancement of parameter estimators via dynamic regressor extension and mixing," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3546–3550, 2017.

[12] P. Thomas Fletcher, "Geodesic regression and the theory of least squares on Riemannian manifolds," *International journal of computer vision*, vol. 105, pp. 171–185, 2013.

[13] Y. Karayiannidis, C. Smith, F. E. Vina, and D. Kragic, "Online contact point estimation for uncalibrated tool use," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5 2014.

[14] C. Sloth and I. Iturrate, "Simultaneous contact point and surface normal estimation during soft finger contact," in *2021 20th International Conference on Advanced Robotics (ICAR)*, IEEE, 12 2021.

[15] N. Boumal, *Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.

[16] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a MATLAB toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, no. 42, pp. 1455–1459, 2014.

[17] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. USA: Prentice-Hall, Inc., 1989.

[18] S. T. Smith, "Optimization techniques on Riemannian manifolds," *Fields Institute Communications*, vol. 3, 1994.

[19] P. M. Lion, "Rapid identification of linear and nonlinear systems.," *AIAA Journal*, vol. 5, pp. 1835–1842, 10 1967.

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Force Control*, pp. 363–405. London: Springer London, 2009.

[21] F. Caccavale, B. Siciliano, and L. Villani, "The role of euler parameters in robot control," *Asian journal of control*, vol. 1, no. 1, pp. 25–34, 1999.

[22] G. Kreisselmeier, "Adaptive observers with exponential rate of convergence," *IEEE Transactions on Automatic Control*, vol. 22, pp. 2–8, 2 1977.

[23] S. Aranovskiy, R. Ushirobira, M. Korotina, and A. Vedyakov, "On preserving-excitation properties of kreisselmeier's regressor extension scheme," *IEEE Transactions on Automatic Control*, vol. 68, pp. 1296–1302, 2 2023.