Proceedings of the ASME 2014 Dynamic Systems and Control Conference
DSCC2014
October 22-24, 2014, San Antonio, TX, USA

DRAFT: ADAPTIVE MISMATCH COMPENSATION FOR CORIOLIS VIBRATORY GYROSCOPES WITH IMPROVED CONVERGENCE RATE

Fu Zhang*
Department of Mechanical Engineering
University of California at Berkeley
Berkeley, California 94720
Email: shallen@berkeley.edu

Ehsan Keikha
Behrooz Shahsavari
Roberto Horowitz
Department of Mechanical Engineering
University of California at Berkeley
Berkeley, California 94720

ABSTRACT

This paper presents an online adaptive algorithm to compensate damping and stiffness frequency mismatches in Coriolis Vibratory Gyroscopes (CVG). The proposed adaptive compensator consists of a least square estimator that estimates the damping and frequency mismatches, and an online compensator that corrects the mismatches. In order to improve the convergence rate of the adaptive compensator, we introduce a calibration phase where we learn the relations between unknown parameters (i.e. mismatches, rotation rate and rotation angle). The calibration results show that the unknown parameters are on a hyperplane. When the gyroscope is in operation, we project the estimated parameter from the least square estimator in the adaptive compensator onto the learned hyperplane. The projection will reduce the degree of freedom in parameters estimation, thus improving the convergence rate. Simulation results show that utilizing the projection method in the online adaptive compensator will drastically improve the convergence rate and eliminate the rotation angle measurement error.

INTRODUCTION

Coriolis Vibratory Gyroscopes (CVG) are sensing devices that measure the rotation rate and rotation angle of the base where they are mounted. A CVG is usually modeled as a two-dimensional oscillator [1]. As shown in Fig.1, the readout axis, stiffness axis and damping axis are misaligned, as a result of fabrication imperfections.

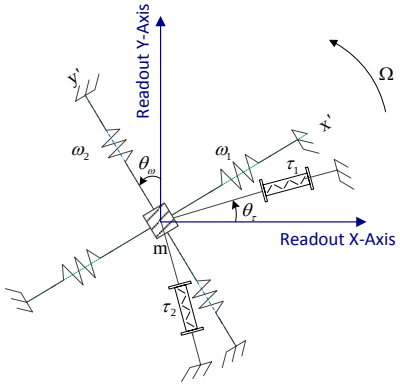

FIGURE 1. A two dimensional oscillator model. m is the effective mass of the gyroscope, Ω is the external input rotation rate, ω_1 and ω_2 are the two resonance frequencies along two axises respectively, τ_1 and τ_2 are the two time constants along two axises respectively, θ_ω is the azimuthal angle between the readout axis and stiffness axis and θ_τ is the azimuthal angle between the readout axis and damping axis.

Utilizing the Newton's second law and projecting spring forces and damping forces onto readout axises, we obtained the governing equations of the readout signals (i.e. x and y),.

*Address all correspondence to this author.

$$\ddot{x} - 2\kappa\Omega\dot{y} - \kappa\dot{\Omega}y + \frac{2}{\tau}\dot{x} + \Delta(\frac{1}{\tau})(\dot{x}\cos 2\theta_\tau + \dot{y}\sin 2\theta_\tau) + (\omega^2 - \kappa^2\Omega^2)x - \omega\Delta\omega(x\cos 2\theta_\omega + y\sin 2\theta_\omega) = \frac{f_x}{m} \quad (1)$$

$$\ddot{y} + 2\kappa\Omega\dot{x} + \kappa\dot{\Omega}x + \frac{2}{\tau}\dot{y} - \Delta(\frac{1}{\tau})(-\dot{x}\sin 2\theta_\tau + \dot{y}\cos 2\theta_\tau) + (\omega^2 - \kappa^2\Omega^2)y - \omega\Delta\omega(x\sin 2\theta_\omega - y\cos 2\theta_\omega) = \frac{f_y}{m} \quad (2)$$

where f_x, f_y are drive forces being applied along x and y axis respectively, $\omega^2 = \frac{\omega_1^2+\omega_2^2}{2}, \omega\Delta\omega = \frac{\omega_1^2-\omega_2^2}{2}, \frac{2}{\tau} = \frac{1}{\tau_1} + \frac{1}{\tau_2}, \Delta(\frac{1}{\tau}) = \frac{1}{\tau_1} - \frac{1}{\tau_2}$; Term $2\kappa\Omega\dot{y}$ and $2\kappa\Omega\dot{x}$ are called coriolis forces, which are proportional to the external rotation rate by the gain factor κ. The coriolis effect is the key principle of how a CVG is able to measure the rotation rate.

[2] presents an algorithm that directly utilizes the coriolis effect to measure the rotation rate. Due to its simplicity and low computational cost, it has been widely used in low-cost commercial applications. However, it requires calibrating gyroscopes' parameters ahead of operation. And the rotation rate measurement precision is extremely sensitive to the calibration precision, especially when the gyroscope is underdamped (i.e. small $\frac{1}{\tau}$). Moreover, when ω_1 considerably differentiates from ω_2, the algorithm's Signal Noise Ratio (SNR) will degrade dramatically. Finally, it does not directly measure the rotation angle. As a result, it barely meets applications where rotation angle needs to be measured precisely

A more sophisticated algorithm is therefore required to control the vibration of CVGs in a known trajectory that is directly related to the external rotation angle. The desired vibration trajectory of a CVG is shown in Fig.2, where the CVG's vibration consists of oscillation and precession. The oscillation is at the resonance frequency ω, with a desired energy level a. The precession is at the rate of $-\kappa\Omega$, with zero quadrature. Then demodulation techniques can be used to separate the precession motion from oscillation motion. Once knowing the precession angle (i.e. $-\kappa\int_0^t \Omega(\tau)d\tau$), we are able to measure the external rotation angle (i.e. $\int_0^t \Omega(\tau)d\tau$) by calibrating the constant κ offline.

In order to achieve this control goal, [3] proposed a method of averaging. It decomposes the control goal into three subgoals [4]: energy control, quadrature control and phase-locked loop (PLL). In the energy control loop, it attempts to compensate for the system damping and maintain system's vibration at a desired energy level. In the quadrature loop, it attempts to minimize system's quadrature and In the PLL, it attempts to

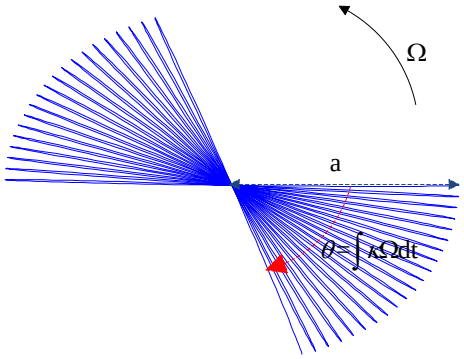

FIGURE 2. The desired trajectory of a CVG.

lock the resonance frequency (ω) in a online fashion and then generate a reference signal that is later used in the demodulation module to demodulate the energy, quadrature, phase error and precession angle. In each control loop, it utilizes a basic PI controller to achieve its subgoal. Each PI controller takes its controlled variable (i.e. energy, quadrature and phase error) and computes the correspond control action. Then the three control actions are synthesized into drive forces f_x and f_y, thus forming a closed loop control systems. This method is easy to understand since its control loops and controlled variables are directly related to the real physical parameters. Because demodulation and PI control are both easy to implement, it has a very low computational cost. Moreover, it does control the gyroscope to vibrate along the desired trajectory when the gyroscope has little mismatch. As a result, it has been widely used in martial and aerospace gyroscopes. However, [5] shows that when system's mismatches are significant, they will cause several fatal problems like non-zero quadrature, precessing at a wrong rate or even failure to precess.

To eliminate the affection from damping and stiffness frequency mismatches, several approaches have been proposed. Among them, the first one is eliminating the damping and stiffness frequency mismatches to a level that does not significantly affect gyro performance during the manufacturing process. However, such manufacturing process is rather sophisticated, requires iterative polishing and extremely fine fabrication. This will drastically increase the manufacturing cost and not feasible in a standard product line. The second approach is to compensate for the mismatches via electrostatic spring softening and trimming during a calibration phase [6]. However, these forms of calibration are time consuming, require human involvement and cannot be performed while the gyro is in operation.The third option is compensating for the mismatches by improving control algorithms. In [7] and [8], Lyapunov based method adaptive controllers were proposed and derived. Simulation

results were supplied to show its efficiency on continuous time gyroscope models. However, if the controller is implemented on systems with relatively low sampling rate, its performance will be degraded dramatically. And this type of controller needs to access velocities, which are not measurable in a practical gyroscope. Therefore, this type of controller has not been used in industry yet.

On top of these work, [5] proposed a more practical compensation algorithm to compensate the mismatches. The proposed compensator is running on top of the method of averaging as an add-on feed-forward compensator. PI controllers in the method of averaging provide a baseline controllers with enough robustness while the adaptive compensator is dedicated to compensating the mismatches. The combination of these two can eliminate mismatches drastically. Simulation results show that by adding the adaptive mismatch compensator onto the method of averaging, the quadrature can be eliminated drastically compared with the basic method of averaging and the gyro will precess at the correct rate (i.e. $-\kappa\Omega$). However, due to its low convergence rate, the compensator has a rather long transient response. As a result, the precession angle has a considerable DC bias compared with the real external rotation angle.

This paper attempts to reduce the transient time in the adaptive compensator proposed in [5] by introducing an offline calibration phase ahead of the gyroscope operation. During the calibration phase, the adaptive mismatch compensator estimates the damping and stiffness frequency mismatches under given external rotation rate. Iterating this process over different external rotation rate, we are able to learn the relations between mismatches and external rotation rate. The learned relations prove to be linear, thus defining a hyperplane in terms of damping mismatch, stiffness frequency mismatch and rotation rate. When the gyroscope is in operation, the adaptive mismatch compensator estimates all the parameters (i.e. damping mismatches, stiffness frequency mismatches and rotation rate) using least square method. The estimated parameters are then projected to the learned hyperplane. The action of projection will reduce the Degree Of Freedom (DOF) in parameters' estimation, thus improving the convergence rate. Simulation study shows that by using projection in the online adaptive compensator, gyroscopes' performance will be improved dramatically.

This paper is organized as below: we first review the adaptive compensator proposed in [5], analyze its convergence rate and performance. Then we introduce an offline calibration algorithm that learns the relation between mismatches and external rotation rate. Afterwards, we will show how to cooperate the offline calibrations into the online adaptive compensator. Finally, simulation study will be supplied to validate its efficiency.

REVIEW AND ANALYSIS OF ADAPTIVE MISMATCH COMPENSATOR

The adaptive mismatch compensator proposed in [5] runs two independent compensators. One of them is in quadrature loop and the other one is in angular precession loop. Each compensator consists of an estimator, which is estimating the effective mismatches using least square method, and a compensator, which is correcting the mismatches.

Since the compensator in quadrature loop converges much faster than the one in angular precession loop, the quadrature compensator shows little effect on the precession angle estimation. Therefore, we only consider how the angular precession compensator behaves from now on.

Dynamics of Angular Precession Loop

The dynamics of angular precession loop is [3]:

$$\dot{\theta} = -\kappa\Omega + \frac{1}{2}\Delta\left(\frac{1}{\tau}\right)\sin 2(\theta-\theta_\tau) - \frac{f_{q_s}}{2\omega\sqrt{E}} \tag{3}$$

where θ is the precession angle, f_{q_s} is the related control force. In the method of averaging, f_{q_s} is set to be zero in order to let the gyroscope free to precess at the rate $-\kappa\Omega$. E is the system energy level. It worths mentioning that θ and E can be measured by demodulating the sensed signals x and y.

The discretized dynamics are therefore

$$\theta(k+1) = \theta(k) - \kappa\Omega t_s + a_\theta\cos 2\theta(k) + b_\theta\sin 2\theta(k) + f_\theta \tag{4}$$

where t_s is the sampling time, $a_\theta = -\frac{1}{2}\Delta\left(\frac{1}{\tau}\right)t_s\sin 2\theta_\tau, b_\theta = \frac{1}{2}\Delta\left(\frac{1}{\tau}\right)t_s\cos 2\theta_\tau$ are the damping mismatches to be estimated. The feed-forward compensation action to be designed is $f_{q_s} = -2\omega\sqrt{E}f_\theta$.

Design of The Adaptive Mismatch Compensator in Angular Precession Loop

Utilizing the fact that mismatches (i.e.a_θ, b_θ) disturb precession angle dynamics as sinusoidal functions of the precession angle, the adaptive mismatch compensator estimates the mismatches using least square estimation method.

Let $\Theta = [\kappa\Omega t_s, a_\theta, b_\theta]^T$ be the unknown parameter vector to estimate, $\hat{\Theta}(k) = \left[\kappa\hat{\Omega}(k)t_s, \hat{a}_\theta(k), \hat{b}_\theta(k)\right]^T$ be the estimated parameters at step k, $\Psi(k) := [-1, \cos 2\theta(k-1), \sin 2\theta(k-1)]^T$ be the known regressor. The algorithm is:

Setup current step is $k+1$, the estimated parameter at previous step is $\hat{\Theta}(k)$, the precession angle up to previous step is $\theta(k)$.

Step 1 Update internal dynamics: $\hat{\theta}(k+1) = \theta(k) - \kappa\hat{\Omega}(k)t_s$;
Step 2 Demodulate the precession angle at current step $\theta(k+1)$;
Step 3 Compute the A-priori error:$e^o(k+1) = \theta(k+1) - \hat{\theta}(k+1)$;
Step 4 Form the regressor: $\Psi(k+1) = [-1, \cos 2\theta(k), \sin 2\theta(k)]^T$;
Step 5 Update parameter estimation by calling the standard recursive least square estimator routine: $\hat{\Theta}(k+1) = RLSE(\hat{\Theta}(k), e^o(k+1), \Psi(k+1))$;
Step 6 Compute the compensation actuation: $f_\theta(k+1) = -\hat{a}_\theta(k+1)\cos 2\theta(k+1) - \hat{b}_\theta(k+1)\sin 2\theta(k+1)$;
Step 7 Upon receiving the next sample, increase k by one and go back to step 1.

Convergence Analysis of The Adaptive Mismatch Compensator

By simple derivation, it can be shown that the adaptive compensation algorithm in above subsection has

$$e^o(k) = \tilde{\Theta}(k-1)^T\Psi(k) \tag{5}$$

where $\tilde{\Theta}(k-1) = \Theta - \hat{\Theta}(k-1)$ is the parameter estimation error.

This form of error follows the same form as the A-priori error in recursive least square estimation problems. Therefore, we can analyze its convergence performance by checking its persistence of excitation, which is defined as

$$PE = rank\left(\lim_{N\to\infty}\frac{\sum_{k=0}^{N}\Psi(k)\Psi(k)^T}{N}\right) \tag{6}$$

In our problem, it is easy to see that $PE = 3$. It means the least square estimator will converge when the number of estimated parameters are no more than 3. Therefore, the $\hat{\Omega}, \hat{a}_\theta$ and $\hat{b}_\theta$ will converge to their real values.

However, since persistence of excitation is no more than the number of estimated parameters, the parameters converge in the rate of $\kappa\Omega$, which is rather small in most applications. As a result, the parameters converge extremely slowly, causing a long transient time. The long transient response will increase the DC bias between the precession angle and the real external rotation angle.

In next sections, we attempt to find some relations between the unknown parameters during the calibration phase. The found relations can give more excitation to the adaptive compensator, thus improving its convergence rate.

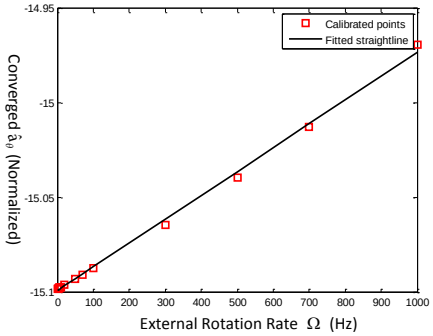

FIGURE 3. Simulated a_θ as a function of Ω during the calibration phase. The simulation setup is in Appendix A.

OFFLINE CALIBRATION PHASE

During the calibration phase, we want to learn the relation between the unknown parameters (i.e. a_θ, b_θ and Ω). To achieve this goal, we can calibrate the gyroscope on a rotation table. Then perform below steps:

Step 1 Set the rotation rate of the rotation table;
Step 2 Run the adaptive compensator with a known Ω. This means the unknown parameters becomes $\Theta = [a_\theta, b_\theta]^T$, the regressor reduces to be $\Psi(k) = [\cos 2\theta(k-1), \sin 2\theta(k-1)]^T$ and the internal dynamics update is $\hat{\theta}(k+1) = \theta(k) - \kappa\Omega t_s$;
Step 3 After convergence, record the triple $(\Omega, \hat{a}_\theta, \hat{b}_\theta)$;
Step 4 set the rotation rate to be another value and go back to Step 2 until the number of calibrated rotation rate exceeds the maximal number predefined.

Running above algorithm, we get the relation between a_θ, b_θ and Ω shown in Fig.3 and Fig.4. where we can find that both a_θ and b_θ are linearly dependent on Ω. We guess that this linear relation is not a coincidence, but coming from the nature of gyroscopes' dynamics discretization. Fitting the measured data using a first order polynomial, we get

$$a_\theta = a_1\Omega + b_1 \tag{7}$$
$$b_\theta = a_2\Omega + b_2 \tag{8}$$

Or equivalently

$$A \cdot \Theta = b \tag{9}$$

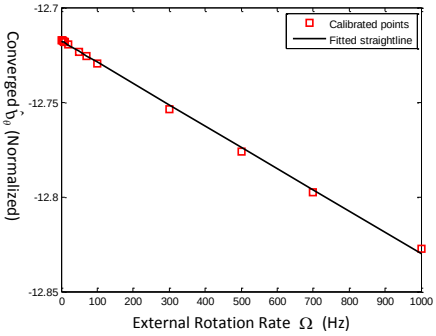

FIGURE 4. Simulated b_θ as a function of Ω during the calibration phase. The simulation setup is in Appendix A.

where $\Theta = [\kappa t_s \Omega, a_\theta, b_\theta]^T; A = \begin{bmatrix} \frac{a_1}{\kappa t_s} & -1 & 0 \\ \frac{a_2}{\kappa t_s} & 0 & -1 \end{bmatrix}; b = \begin{bmatrix} -b_1 \\ -b_2 \end{bmatrix}$; Eqn.(9) defines a very nice linear constraint on the unknown parameters. This constraint can later be used in the online adaptive compensator to improve its convergence rate.

ADAPTIVE MISMATCH COMPENSATOR WITH IMPROVED CONVERGENCE RATE

During the calibration phase, we learned a nice linear relation between the unknown parameters. We might as well utilize this auxiliary information in the online adaptive compensator to improve its convergence rate.

Adaptive Mismatch Compensator With Hard Projection

The adaptive mismatch compensator in [5] uses a standard least square method to estimate the unknown parameters Θ. Analysis from last section shows that this standard least square method converges very slowly in our problem. Meanwhile, we already know that the unknown parameter are on the hyperplane defined by Eqn.(9). One way to combine these two together is projecting the estimated parameter from the standard least square estimator onto the learned hyperplane.

Definition: Projecting a point y onto a hyperplane defined as $\{x|Ax = b\}$ means solving below optimization problem

$$x^* = \arg\min_x \frac{1}{2}\|x-y\|_2^2$$
$$s.t.\ Ax - b = 0$$

where AA^* is invertible.

Solution of above optimization can be easily found by using the Karush-Kuhn-Tucker (KKT) conditions:

$$x^* = [I - A^*(AA^*)^{-1}A]y + A^*(AA^*)^{-1}b \quad (10)$$
$$= Ky + c \quad (11)$$

where $K = I - A^*(AA^*)^{-1}A$ and $c = A^*(AA^*)^{-1}b$. K and c can be computed offline after the calibration phase to save online computation burden. As a result, K is a 3 by 3 matrix while c is a 3 by 1 vector. The projection only uses very few computations.

Therefore, the algorithm of adaptive mismatch compensator with hard projection is

Setup K and c were computed during the calibration phase. Current step is $k+1$, the estimated parameters at precious step is $\hat{\Theta}(k)$, the precession angle up to previous step is $\theta(k)$.

Step 1 Update internal dynamics: $\hat{\theta}(k+1) = \theta(k) - \kappa\hat{\Omega}(k)t_s$;
Step 2 Demodulate the precession angle at current step $\theta(k+1)$;
Step 3 Compute the A-priori error:$e^o(k+1) = \theta(k+1) - \hat{\theta}(k+1)$;
Step 4 Form the regressor: $\Psi(k+1) := [-1, \cos 2\theta(k), \sin 2\theta(k)]^T$;
Step 5 Update the parameter estimation by calling the standard recursive least square estimator routine: $\hat{\Theta}^o(k+1) = RLSE(\hat{\Theta}(k), e^o(k+1), \Psi(k+1))$;
Step 6 Project the a-priori estimation onto the calibrated hyperplane: $\hat{\Theta}(k+1) = K \cdot \hat{\Theta}^o(k+1) + c$
Step 7 Compute the compensation actuation: $f_\theta(k+1) = -\hat{a}_\theta(k+1)\cos 2\theta(k+1) - \hat{b}_\theta(k+1)\sin 2\theta(k+1)$;
Step 8 Upon receiving the next sample, increase k by one and go back to step 1.

Since the projection operator is non-expansive, adding the projection in the least square estimator does not alter its convergence. Since the projection reduces the DOF of the estimated parameters, it does improve the convergence rate.

Adaptive Mismatch Compensator With Soft Projection

One potential risk of using the hard projection in the adaptive mismatch compensator is that the calibrated hyperplane may not be so accurate due to the calibration error. Brutally projecting the estimated parameter onto the inaccurate hyperplane helps the convergence rate at the beginning of the least square estimation. However, the estimated parameters will never converge to its true values, but their projections on the hyperplane if the true values are not exactly on the hyperplane. This will introduce steady errors in the parameters estimation as well as gyroscopes' angular precession.

To avoid this risk, we remove the constraints into the objective function and penalize it with a designed weight. We call it soft projection.

Definition: Softly projecting a point y onto a hyperplane defined as $\{x|Ax = b\}$ means solving below optimization problem

$$x^* = \arg\min_x \frac{1}{2}\|x-y\|_2^2 + \gamma \cdot r(Ax-b)$$

where AA^* is invertible. $r(\cdot)$ is a non-negative and convex function. γ is a adjustable constant that weight the online least square estimation and the offline calibration. This is also known as a proximity operator, denoted as $x = \text{prox}_r(y)$.

This is an unconstrained optimization problem. In our problem, a simple choice for $r(\cdot)$ is $r(x) = \frac{1}{2}\|x\|_2^2$, which is convex, non-negative and differentiable. By setting objective function's differentiation zero, we can get the solution

$$x^* = (I + \gamma A^* A)^{-1}(y + \gamma A^* b) \tag{12}$$

By adjusting γ, we put different faith on the hyperplane learned from offline calibration phase. When the offline calibration makes no sense, we set $\gamma = 0$. Then $x^* = y$, which means the estimated parameter directly comes from the least square estimation.

When the offline calibration is very accurate and we would like to use it to guide the online estimation, then we set $\gamma \to +\infty$.

$$x^* = \lim_{\gamma \to +\infty} (I + \gamma A^* A)^{-1}(y + \gamma A^* b) \tag{13}$$

As γ approaches to infinity, above solution goes to the hard projection solution because when $\gamma = +\infty$, in order to prevent the objective function from infinity, $r(Ax-b) = 0$, as a result, $Ax - b = 0$ has to hold.

When the offline calibration is reasonable but not very accurate, which is the most usual case in practice. We set $\gamma \in (0, +\infty)$. The more accurate the offline calibration is, the larger γ we use. To save computations, let $K' = (I + \gamma A^* A)^{-1}; c' = \gamma \cdot (I + \gamma A^* A)^{-1} A^* b$, which can be computed offline during the calibration phase, then

$$x^* = K' y + c' \tag{14}$$

Therefore, the algorithm of adaptive mismatch compensator with soft projection is

Setup K' and c' were computed during the calibration phase. Current step is $k+1$, the estimated parameters at precious step is $\hat{\Theta}(k)$, the precession angle up to previous step is $\theta(k)$.

Step 1 Update internal dynamics: $\hat{\theta}(k+1) = \theta(k) - \kappa\hat{\Omega}(k)t_s$;
Step 2 Demodulate the precession angle at current step $\theta(k+1)$;
Step 3 Compute the A-priori error:$e^o(k+1) = \theta(k+1) - \hat{\theta}(k+1)$;
Step 4 Form the regressor: $\Psi(k+1) := [-1, \cos 2\theta(k), \sin 2\theta(k)]^T$;
Step 5 Update parameter estimation by calling the standard recursive least square estimator routine: $\hat{\Theta}^o(k+1) = LSE(\hat{\Theta}(k), e^o(k+1), \Psi(k+1))$;
Step 6 Correct the a-priori estimation: $\hat{\Theta}(k+1) = K' \cdot \hat{\Theta}^o(k+1) + c'$
Step 7 Compute the compensation actuation: $f_\theta(k+1) = -\hat{a}_\theta(k+1)\cos 2\theta(k+1) - \hat{b}_\theta(k+1)\sin 2\theta(k+1)$;
Step 8 Upon receiving the next sample, increase k by one and go back to step 1.

Since the proximity operator is non-expansive, adding the soft projection onto the standard least square estimator does not alter its convergence. Since the soft projection reduces the DOF of the estimated parameters, it does improve the convergence rate.

γ can also be time varying. At the beginning, we set a large γ to improve the convergence rate. As the parameters approach to their true values, γ decreases to increase the significance of online adaptation on parameter estimation such that parameters will finally converge to its true values. This requires more computations since K', c' have to be updated in real time as γ changes.

SIMULATION RESULTS

In order to verify its efficiency, the proposed adaptive mismatch compensator with hard/soft projection are tested on the simulated gyroscope model with parameters provided in Appendix A.

Efficiency of Hard Projection

Fig.5, Fig.6 and Fig.7 show the comparison of parameters (mismatches a_θ, b_θ and external rotation rate Ω) estimation, between the standard adaptive mismatch compensator and the adaptive mismatch compensator with hard projection. It can be seen that both converges to the same values. But with the hard projection, the compensator converges extremely fast.

Fig.8 shows the precession angle response. It can be seen that due to its low convergence rate, the standard adaptive mismatch compensator has 45 degrees of rotation angle measurement error. By using projection in the compensator, its convergence rate is drastically improved. As a result, the precession angle can follow

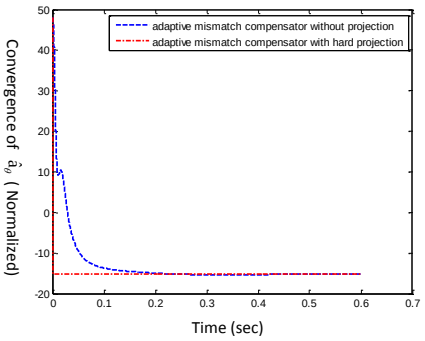

FIGURE 5. Simulated convergence of parameter a_θ, with hard projection on (dash-dot red) and off (dashed blue)

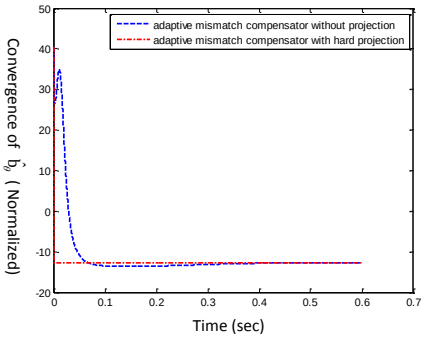

FIGURE 6. Simulated convergence of parameter b_θ, with hard projection on (dash-dot red) and off (dashed blue)

the ground true rotational angle very precisely. A finer plot of the precession angle estimation error is shown in Fig.9, where we can see that the the compensator with projection has maximal 0.005 degree of rotation angle estimation error.

Simulation Study of the Compensator with Soft Projection

The adaptive mismatch compensator with soft projection is tested on the simulated gyroscope as well. Fig.10, Fig.11 and Fig.12 show the convergence of parameters' estimation under different penalty weight (i.e. γ). We can see that both converges to the same values. But by putting less faith (i.e. smaller γ), the convergence rate is lower but gaining more robustness to the offline calibration. Lower convergence rate causes a bigger bias in the rotation angle estimation, as shown in Fig.13. But both cases are better than the standard adaptive mismatch compensator in terms of angle estimation error.

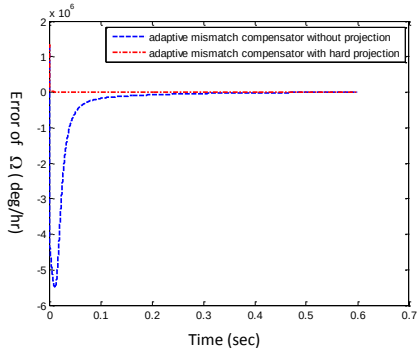

FIGURE 7. Simulated convergence of Ω error, with hard projection on (dash-dot red) and off (dashed blue)

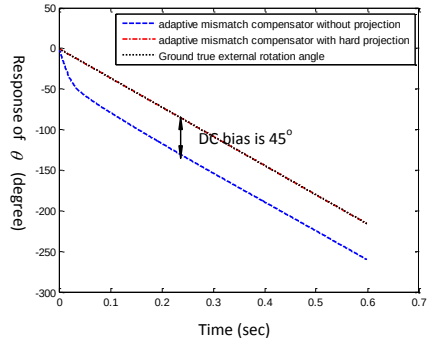

FIGURE 8. Simulated precession angle θ response, with hard projection on (dash-dot red) and off (dashed blue), which is covered by the ground true rotation angle (dotted black)

CONCLUSION

In this paper, we have introduced a projection method to improve the convergence rate in the adaptive mismatch compensator proposed in [5]. This method starts with leaning the hyperplane where the unknown parameters are located on during the calibration phase. Then the learned hyperplane is used as a constraint in the least square estimator. By projecting the estimated parameters from the least square method onto the hyperplane, it reduces the parameters' degree of freedom thus improving the convergence rate drastically. Simulation results are supplied to show that the projection method drastically improved the parameters' convergence rate and eliminated the DC bias in the angle estimation.

Inspired by the projection, we proposed a soft projection by penalizing the constraint in the objective function, instead of brutally projecting onto the hyperplane. This a good way to combine the offline calibration with online estimation.

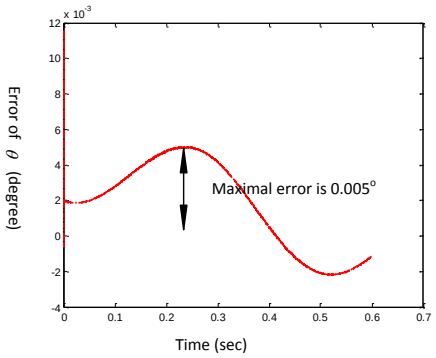

FIGURE 9. Simulated convergence of Ω error, with hard projection on (dash-dot red) and off (dashed blue)

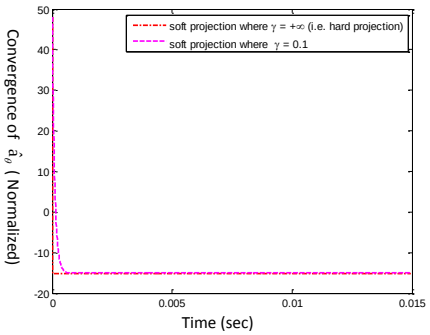

FIGURE 10. Simulated convergence of parameter a_θ

Moreover, the penalty weight (i.e.γ) gives a way to adjust how much significance the offline calibration should guide the online estimation. Simulation results show that this soft projection also improves the convergence rate and angle estimation error.

Future work can be done in several aspects: implementing the proposed algorithm on real gyroscopes; Analyzing the performance of the projection method when rotation rate Ω is time varying; generalizing the projection to nonlinear constraints and other applications.

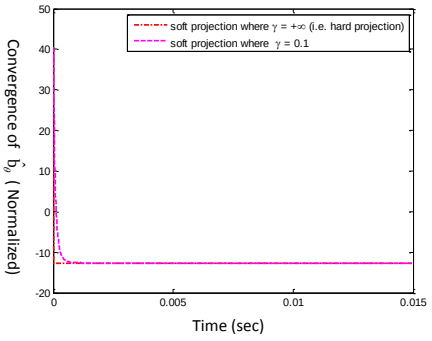

FIGURE 11. Simulated convergence of parameter b_θ

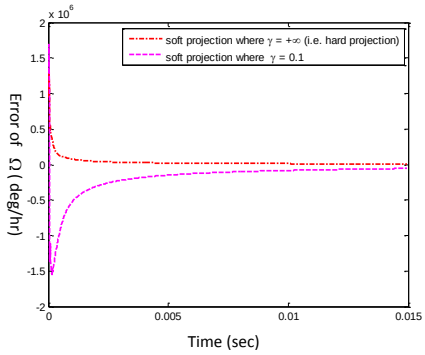

FIGURE 12. Simulated convergence of Ω error

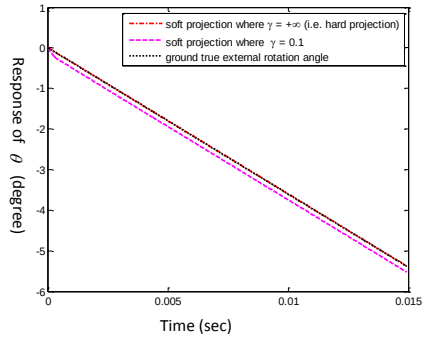

FIGURE 13. Simulated precession angle response

ACKNOWLEDGMENT

This research is supported by the DARPA MRIG program, through a subcontract from Honeywell Aerospace Advanced Technology. We would like to thank Dr. Burgess R. Johnson and all the collaborators from Honeywell for their gyroscope modeling data and continuous feedback on control design.

REFERENCES

[1] Lynch, D. D., 1998. "Coriolis Vibratory Gyro". Symposium Gyro Technology, Stuttgart, Germany.

[2] Acar, C., and Shkel, A., 2008. *MEMS Vibratory Gyroscopes: Structural Approaches to Improve Robustness*. MEMS Ref-

erence Shelf. Springer, Chap. 6, pp. 143–185.
[3] Lynch, D. D., 1995. "Vibratory Gyro Analysis by The Method of Averaging". 2nd St. Petersburg Int. Conf. on Gyroscopic Technology and Navigation, St. Petersburg, Russia, pp. 26–34.
[4] Lynch, D. D., and Matthews, A., 1996. "Dual-Mode Hemisphericcal Resonator Gyro Operation Characteristics". 3rd Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg, Russia, pp. 37–44.
[5] Zhang, F., Keikha, E., Shahsavari, B., and Horowitz, R., 2014. "Adaptive Mismatch Compensation for Vibratory Gyroscopes". The 1st IEEE International Symposium on Inertial Sensors and Systems, pp. 67–70.
[6] Cao, H., Li, H., Ni, Y., and Pan, H., 2014. "Electrostatic force correction for the imperfections of the MEMS gyroscope structure". The 1st IEEE International Symposium on Inertial Sensors and Systems, pp. 71–74.
[7] Park, S., and Horowitz, R., 2001. "Adaptive control for Z-axis MEMS gyroscopes". Vol. 2 of *Proceedings of the 2001*, American Control Conference, IEEE, pp. 1223–1228.
[8] Leland, R. P., 2006. "Adaptive control of a MEMS gyroscope using Lyapunov methods". Vol. 14, Control Systems Technology, IEEE Transactions on, pp. 278–283.

Appendix A: SIMULATION SETUP

Simulations were performed on the gyro system with below parameters:

resonance frequency $\omega = 10,000Hz$;
sampling rate $f_s = 98,000Hz$;
frequency mismatch $\Delta\omega = 5Hz$;
time constant $\tau = 2$sec;
damping mismatch $(\frac{1}{\tau}) = 0.001\text{sec}^{-1}$;
gain factor $\kappa = 1$;
input rotation rate $\Omega = 1Hz$