NIH-PA Author Manuscript
NIH-PA Author Manuscript



Published in final edited form as:
*Conf Proc IEEE Eng Med Biol Soc*. 2011 ; 2011: 4124–4127. doi:10.1109/IEMBS.2011.6091024.

# Efficient Estimation of Time-Varying Intrinsic and Reflex Stiffness

**Daniel Ludvig [Member, IEEE]**,
Sensory Motor Performance Program, Rehabilitation Institute of Chicago, Chicago, IL 60657 USA (phone: 312-238-3381; fax: 312-238-2208; daniel.ludvig@mail.mcgill.ca).

**Eric J. Perreault [Member, IEEE]**, and
Department of Biomedical Engineering and the Department of Physical Medicine and Rehabilitation at Northwestern University, Chicago, IL 60611 USA, and also with Sensory Motor Performance Program, Rehabilitation Institute of Chicago, Chicago, IL 60611 USA (e-perreault@northwestern.edu).

**Robert E. Kearney [IEEE, Fellow]**
Biomedical Engineering Department, McGill University, Montreal, Qc H3A 2B4 Canada (robert.kearney@mcgill.ca).

## Abstract

Dynamic joint stiffness defines the dynamic relationship between the position of the joint and the torque acting about it; hence it is important in the control of movement and posture. Joint stiffness consists of two components: intrinsic stiffness and reflex stiffness. Measuring intrinsic and reflex torques directly is not possible, thus estimating intrinsic and reflex stiffness is challenging. A further complication is that both intrinsic and reflex stiffness vary with joint position and torque. Thus, the measurement of dynamic joint stiffness during movement requires a time-varying algorithm. Recently we described an algorithm to estimate time-varying intrinsic and reflex stiffness and demonstrated its application. This paper describes modifications to that algorithm that significantly improves the accuracy of the estimates it generates while increasing its computational efficiency by a factor of seven.

## I. Introduction

Dynamic joint stiffness defines the dynamic relationship between the position of the joint and the torque acting about it. Joint stiffness consists of two components: intrinsic stiffness, which arises due to the viscoelastic properties of the joint, connective tissue and the inertia of the limb; reflex stiffness, which arises due to the torque produced by the stretch reflex response [1].

Measuring intrinsic and reflex torques directly is not possible in an intact system, thus directly estimating intrinsic and reflex stiffness is not possible. Further complicating the matter is that intrinsic and reflex stiffness appear and change together. A number of different approaches have been used to analytically separate intrinsic and reflex torque, thus producing estimates of the two stiffness components [2–5]. One of these approaches is the parallel-cascade identification algorithm [2]. This algorithm takes advantage of the reflex lag to estimate intrinsic stiffness and then subsequently estimates reflex stiffness. It then iteratively re-estimates each component until the model fails to account for any additional torque variance. Using this algorithm, it has been shown that intrinsic and reflex stiffness vary with the joint position and the activation level of the surrounding muscles [6]. However, this algorithm can only produce estimates under time-invariant (TI) conditions,

thus is not useful for estimating stiffness during time-varying (TV) conditions, such as movement.

A recently published algorithm [7] showed the ability to produce accurate estimates of reflex and intrinsic stiffness under TV conditions. This TV parallel-cascade algorithm (TVPC), was developed by reformulating the TI parallel-cascade algorithm to use an ensemble of input and output realizations, to estimate stiffness at every time point. The algorithm accurately estimated TV changes in simulated systems but was less successful when applied to experimental data. Close inspection of the results showed problems arose when the algorithm failed to iterate following the initial estimate. One possible reason for this was that algorithm was designed to iterate between estimating intrinsic stiffness for all times and estimating reflex stiffness for all times. As a result a bad estimate of intrinsic or reflex stiffness at a single time point would corrupt the estimates for all other time points and cause the iteration to fail.

In this paper we present a modification to the TVPC algorithm. The algorithm was reformulated to iterate between estimating intrinsic stiffness and reflex stiffness for each time point. Once the algorithm converges at that time the algorithm then estimates stiffness for the next point.

This paper is developed as follows: section II presents the modified TV algorithm. Section III presents simulation results showing the improvements made by the modified algorithm. Finally section IV summarizes the findings, and discusses the results.

## II. Time-Varying Parallel-Cascade Algorithm

The TVPC algorithm was presented in great detail in [7], and so will be summarized briefly here.

### A. Estimating Time-Varying Intrinsic Stiffness

Intrinsic stiffness is estimated by computing a TV linear impulse response function (IRF) between the position ($P$) and intrinsic torque ($T$). The TV linear IRF is estimated by solving the following equation

$$\widehat{\widehat{\mathbf{\Phi}}}_{\mathbf{TP}}(i)=\Delta t\widehat{\widehat{\mathbf{\Phi}}}_{\mathbf{PP}}(i)\mathbf{h}_{\mathbf{I}}(i) \quad (1)$$

where

$$\widehat{\mathbf{\Phi}}_{\mathbf{PP}}(i)=\begin{bmatrix} \phi_{PP}(i-M1,0) & \widehat{\phi}_{PP}(i-M1,-1) & \cdots & \widehat{\phi}_{PP}(i-M1,M1-M2) \\ \phi_{PP}(i-M1-1,1) & \widehat{\phi}_{PP}(i-M1-1,0) & \cdots & \widehat{\phi}_{PP}(i-M1,0) \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\phi}_{PP}(i-M2,M2-M1) & \widehat{\phi}_{PP}(i-M2,M2-M1,1) & \cdots & \widehat{\phi}_{PP}(i-M2,0) \end{bmatrix}$$

$$\widehat{\mathbf{\Phi}}_{\mathbf{TP}}(i)=[\widehat{\phi}_{TP}(i,-M1)\widehat{\phi}_{TP}(i,-M1-1)\cdots\widehat{\phi}_{TP}(i,-M2)]^T \mathbf{h}_{\mathbf{I}}(i)=[h_I(i,M1)\cdots h_I(i,0)\cdots h_I(i,M2)]^T$$

$\hat{\phi}_{PP}(i, j)$ is the TV auto-correlation between the position at times $i$ and $j$, $\hat{\phi}_{TP}(i, j)$ is the TV cross-correlation between intrinsic torque at time $i$ and position at time $j$, $h_I$ is the intrinsic stiffness IRF, and $M1$ and $M2$ are the minimum and maximum lag of the intrinsic stiffness IRF.

### B. Estimating Time-Varying Reflex Stiffness

Reflex stiffness is estimated by computing a TV Hammerstein system between the velocity (V) and the reflex torque ($T_R$). To estimate the Hammerstein system, first a TV linear IRF ($h_R$) is found between the velocity and the reflex torque. This is done using Eq. 1 except where the velocity is the input and reflex torque is the output. Once this initial estimate is generated, the static non-linear element and dynamic linear element are estimated iteratively by fixing one of the elements and estimating the other.

Estimating the static non-linear element is done by solving the following least-squares problem

$$\mathbf{T_R}(i)=\mathbf{A}(i)\mathbf{p}(i)+\mathbf{e} \quad (2)$$

where

$$\mathbf{A}(i)=\begin{bmatrix} \Delta t \sum_{j=M1}^{M2} h_R(i,j) & \Delta t \sum_{j=M1}^{M2} h_R(i,j)V(i-j,1) & \Delta t \sum_{j=M1}^{M2} h_R(i,j)V(i-j,1)^N \\ \Delta t \sum_{j=M1}^{M2} h_R(i,j) & \Delta t \sum_{j=M1}^{M2} h_R(i,j)V(i-j,2) & \Delta t \sum_{j=M1}^{M2} h_R(i,j)V(i-j,2)^N \\ \Delta t \sum_{j=M1}^{M2} h_R(i,j) & \Delta t \sum_{j=M1}^{M2} h_R(i,j)V(i-j,R) & \Delta t \sum_{j=M1}^{M2} h_R(i,j)V(i-j,R)^N \end{bmatrix} \mathbf{p}(i)=[p_0(i)p_1(i)\cdots p_N(i)]^T$$

and $p_n$ are the n$^{\text{th}}$ order coefficients of the static non-linear element.

Similarly, the linear dynamic element can be estimated by solving the least squares problem

$$\mathbf{T_R}(i)=\Delta t\mathbf{B}(i)\mathbf{h_R}(i)+\mathbf{e} \quad (3)$$

where

$$\mathbf{B}(i)=\begin{bmatrix} \sum_{n=0}^{N} p_n(i)V(i-M1,1)^n & \cdots & \sum_{n=0}^{N} p_n(i)V(i-M2,1)^n \\ \vdots & \ddots & \vdots \\ \sum_{n=0}^{N} p_n(i)V(i-M1,R)^n & \cdots & \sum_{n=0}^{N} p_n(i)V(i-M2,R)^n \end{bmatrix} \mathbf{h_R}(i)=[h_R(i,M1)\cdots h_R(i,M2)]$$

and $\mathbf{T}_\mathrm{R}$ is the same as in Eq. 2.

### C. Estimating Time-Varying Parallel-Cascade Systems

Estimating TV intrinsic and reflex stiffness would be trivial if intrinsic and reflex torque were directly measurable; however this is not the case. The parallel-cascade (PC) identification algorithm initially estimates intrinsic stiffness using the net torque and then iteratively estimate each component of stiffness using the residual torque. The iteration stops when the model fails to account for any additional torque variance. Fig. 1 (left side) shows a flow diagram of the original version of the TVPC algorithm; it estimates a TV IRF and a TV Hammerstein system, then computes the residuals for all times and all realizations. Similar to the TI case the TVPC ceases to iterate when the model failed to account for any additional torque variance.

The modified version (shown to the right of Fig. 1) of the TVPC algorithm, iteratively estimates stiffness at each time point. Once the model fails to account for any additional

torque at that time point, the algorithm re-starts at the next time point. The procedure is as follows:

1. Intrinsic stiffness, $\hat{\mathbf{h}}_{\mathbf{I}}$ ($i$), is estimated for the 1st time point ($i$) using Eq. 1 where the net torque is used rather than the intrinsic torque.

2. Intrinsic torque, $\hat{\mathbf{T}}_{\mathbf{I}}$ ($i$) at time $i$ for all $R$ realizations is estimated using the convolution

$$\begin{bmatrix} \widehat{T}_I(i,1) \\ \widehat{T}_I(i,2) \\ \vdots \\ \widehat{T}_I(i,R) \end{bmatrix} = \begin{bmatrix} P(i-M2,1) & \cdots & P(i-M1,1) \\ P(i-M2,2) & \cdots & P(i-M1,2) \\ \vdots & \ddots & \vdots \\ P(i-M2,R) & \cdots & P(i-M1,R) \end{bmatrix} * \begin{bmatrix} \widehat{h}_I(i,M1) \\ \vdots \\ \widehat{h}_I(i,M1) \end{bmatrix} \quad (4)$$

3. The intrinsic residual torque at time $i$, $\hat{\mathbf{T}}_{\mathbf{IR}}$ ($i$) is estimated as

$$\begin{bmatrix} \widehat{T}_{IR}(i,1) \\ \widehat{T}_{IR}(i,2) \\ \vdots \\ \widehat{T}_{IR}(i,R) \end{bmatrix} = \begin{bmatrix} T(i,1) \\ T(i,2) \\ \vdots \\ T(i,R) \end{bmatrix} - \begin{bmatrix} \widehat{T}_I(i,1) \\ \widehat{T}_I(i,2) \\ \vdots \\ \widehat{T}_I(i,R) \end{bmatrix} \quad (5)$$

4. The static non-linear element and dynamic linear element of the reflex stiffness are estimated at time $i$, using Eqs. 2 and 3, using the intrinsic residual torque instead of the reflex torque.

5. The reflex torque at time $i$, $\hat{\mathbf{T}}_{\mathbf{R}}$ ($i$) is estimated by

$$\widehat{\mathbf{T}}_{\mathbf{R}}(i) = \Delta t \widehat{\mathbf{B}}(i) \widehat{\mathbf{h}}_{\mathbf{R}}(i) \quad (6)$$

where $\hat{\mathbf{B}}(i)$ and $\hat{\mathbf{h}}_{\mathbf{R}}(i)$ are the same as in Eq. 3.

6. The reflex residual torque at time $i$, $\hat{\mathbf{T}}_{\mathbf{RR}}$ ($i$), is estimated as in Eq. 5, except replacing $\hat{\mathbf{T}}_{\mathbf{IR}}$ ($i$) with $\hat{\mathbf{T}}_{\mathbf{RR}}$ ($i$) and $\hat{\mathbf{T}}_{\mathbf{I}}$ ($i$) with $\hat{\mathbf{T}}_{\mathbf{R}}$ ($i$).

7. The total predicted torque at time is computed as

$$\widehat{\mathbf{T}}(i) = \widehat{\mathbf{T}}_{\mathbf{I}}(i) + \widehat{\mathbf{T}}_{\mathbf{R}}(i) \quad (7)$$

8. The quality of identification is evaluated in terms of variance accounted for between the observed torque at time $i$, $\mathbf{T}(i)$, and the predicted torque at time $i$, $\hat{\mathbf{T}}(i)$.

$$\%VAF = 100\left(1 - \frac{\mathrm{var}(\mathbf{T}(i) - \widehat{\mathbf{T}}(i)}{\mathrm{var}(\mathbf{T}(i))}\right) \quad (8)$$

9. If %VAF is larger than the previous iteration then steps 1–8 are repeated except $\hat{\mathbf{T}}_{\mathbf{RR}}$ ($i$) is used as the output of the identification of intrinsic stiffness in step 1.

10. Steps 1–9 are repeated for every time point, until PC models are obtained for all time points of interest.

Both iterative loops in the algorithm require initial estimates prior to the start of the loop. A better initial estimate will result in the algorithm requiring less iterations, and hence less computation time. In step 1), the initial estimation of intrinsic stiffness at each time point

assumed that reflex torque was zero. Instead, the initial estimate of reflex torque is computed using the reflex stiffness from the previous time point. Similarly, for the Hammerstein estimation in step 4), the initial estimate of the dynamic linear element was computed by finding a linear IRF between the velocity and the reflex torque. Rather, the linear IRF computed in the previous PC iteration is used as the initial estimate.

Another step that was performed to improve computational efficiency of the algorithm was the elimination of redundant computations. For example, the matrix $\hat{\mathbf{\Phi}}_{\mathbf{PP}}(i)$, is the same in each iteration at any given time point, thus this matrix is computed once, stored and reused.

## III. Simulation Studies

Simulations were performed to evaluate the performance of the modified algorithm as compared to the original version.

### A. Simulation Methods

Fig 2. Shows a schematic of the TV model used for the simulations. Simulations of TV ankle stiffness were run using Simulink (The Mathworks inc.).

Intrinsic stiffness was simulated as a $2^{nd}$ order differential equation; reflex stiffness was simulated as a differentiator, a delay of 40 ms, a half-wave rectifier and a $2^{nd}$ order low-pass filter. The gain of reflex stiffness was varied with time; it ramped up and down between 10 and 30 Nm/rad/s with a period of 4 s, and is shown in Fig. 3A.

The position was simulated as a pseudo-random binary sequence (PRBS) with a switch rate of 150 ms and an amplitude of 0.03 rad. Gaussian white noise was added to the output; the variance of the noise increased with time as shown in Fig. 3B. The noise variance was made TV so that time points of both high and low signal-to-noise ratio (SNR) existed.

1000 realizations of data were generated by running the simulations with the same system parameters, but different realizations of position and noise signals. Simulations were run at 1 kHz for 13 seconds, and data was decimated to 200 Hz prior to analysis.

The TVPC algorithm—the original and modified versions—were run on the ensemble of data. To assess the performance of the algorithms, the %VAFs between the estimated torque and simulated torque were computed for total, intrinsic and reflex components as in Eq. 8.

To assess the improvements in computational performance, the time taken to run the algorithm on 1 s of data was computed with and without the better initial estimates and the redundant computations.

### B. Simulation Results

Fig. 3A–C show the simulated reflex stiffness gain, the simulated noise variance and the resultant SNR. Fig. 3D shows that the algorithm estimated both intrinsic and reflex stiffness accurately until about 12 s, corresponding to a time with low SNR.

Fig. 4 compares the performance of the original algorithm with that of the modified version. It is evident (Fig. 4A–C) that new version osutperformed the algorithm at low SNRs (< 5 dB SNR); the original algorithm failed to account for any torque variance, whereas the modified algorithm accounted for almost all the variance of the intrinsic and total torque and 80–90% of the variance of the reflex torque. Even at higher levels of SNR (Fig. 4D–F), the modified algorithm outperformed the original version. The new version accounted for 98–100%, 99% and 95–100% of the total, intrinsic and reflex torques respectively, while the original

algorithm only accounted for 96–98%, 90–98% and 70–95% of the three torques respectively. The discrepancy was greatest at around 17 and 27 db SNR, which corresponded to times in the simulations when reflex stiffness gain was quite high.

Table 1 shows the total computation time, the number of PC loops and time per PC loop for three versions of the modified TVPC algorithm. Ver. 1 is without either of the computational improvements, Ver. 2 is after eliminating redundant computations and Ver. 3 is after eliminating both redundant computations and using better initial estimates. Eliminating redundant computations decreased time per PC loop by more than twofold, resulting in a similar decrease in computational time. Using better initial estimates did not alter time per PC loop significantly, but did drastically reduce the number of PC loop iterations, thus greatly reducing the total time taken.

## IV. Discussion

This paper presents modifications to a previously published algorithm for TV identification of intrinsic and reflex stiffness. Simulation studies showed that this modified algorithm produces estimates that are more accurate, especially in low SNR conditions.

The main motivation for improving the algorithm was its poor performance with actual data. The original algorithm worked well with simulated data, but its performance dropped when working with experimental data. One potential reason is the variability of the noise in experimental data. A flaw in the original algorithm was that if the stiffness at even one time point was misestimated, then the algorithm failed to iterate. This mis-estimation is generally caused by low SNR, and it is more likely that there may be time points in the experimental data that SNR is low. The simulation studies in this paper were designed to demonstrate this specifically. Even at relatively high SNR (10–30 dB) the modified algorithm performed significantly better than the original, because the original algorithm was not able to iterate to the correct values.

One drawback of this increased algorithm iterations was the increase in computation time. To counter this we reduced the number of iterations needed for the algorithm to converge. We achieved this by using better initial estimates prior to entering the iterative loops. We also were able to reduce computational time by eliminating redundant computations. Combining these computation time saving operations with careful programming allowed us to reduce computational time by sevenfold. Computational performance may further be improved by designing the algorithm to take advantage of the parallel-processing available in modern day computers.

## Acknowledgments

This work was supported by Canadian Institutes of Health Research and the National Institutes of Health (grant R01 NS053813).

## References

1. Kearney RE, Hunter IW. System identification of human joint dynamics. Crit Rev Biomed Eng. 1990; vol. 18:55–87. [PubMed: 2204515]
2. Kearney RE, et al. Identification of intrinsic and reflex contributions to human ankle stiffness dynamics. IEEE Trans Biomed Eng. 1997 Jun.vol. 44:493–504. [PubMed: 9151483]
3. Ludvig D, Kearney RE. Real-time estimation of intrinsic and reflex stiffness. IEEE Trans Biomed Eng. 2007 Oct.vol. 54:1875–1884. [PubMed: 17926686]

4. Zhang LQ, Rymer WZ. Simultaneous and nonlinear identification of mechanical and reflex properties of human elbow joint muscles. IEEE Trans Biomed Eng. 1997 Dec.vol. 44:1192–1209. [PubMed: 9401219]
5. Perreault EJ, et al. Estimation of intrinsic and reflex contributions to muscle dynamics: a modeling study. IEEE Trans Biomed Eng. 2000 Nov.vol. 47:1413–1421. [PubMed: 11077734]
6. Mirbagheri MM, et al. Intrinsic and reflex contributions to human ankle stiffness: variation with activation level and position. Exp Brain Res. 2000 Dec.vol. 135:423–436. [PubMed: 11156307]
7. Ludvig D, et al. Identification of Time-Varying Intrinsic and Reflex Joint Stiffness. IEEE Trans Biomed Eng. 2011 Jun.vol. 58:1715–1723. [PubMed: 21317071]

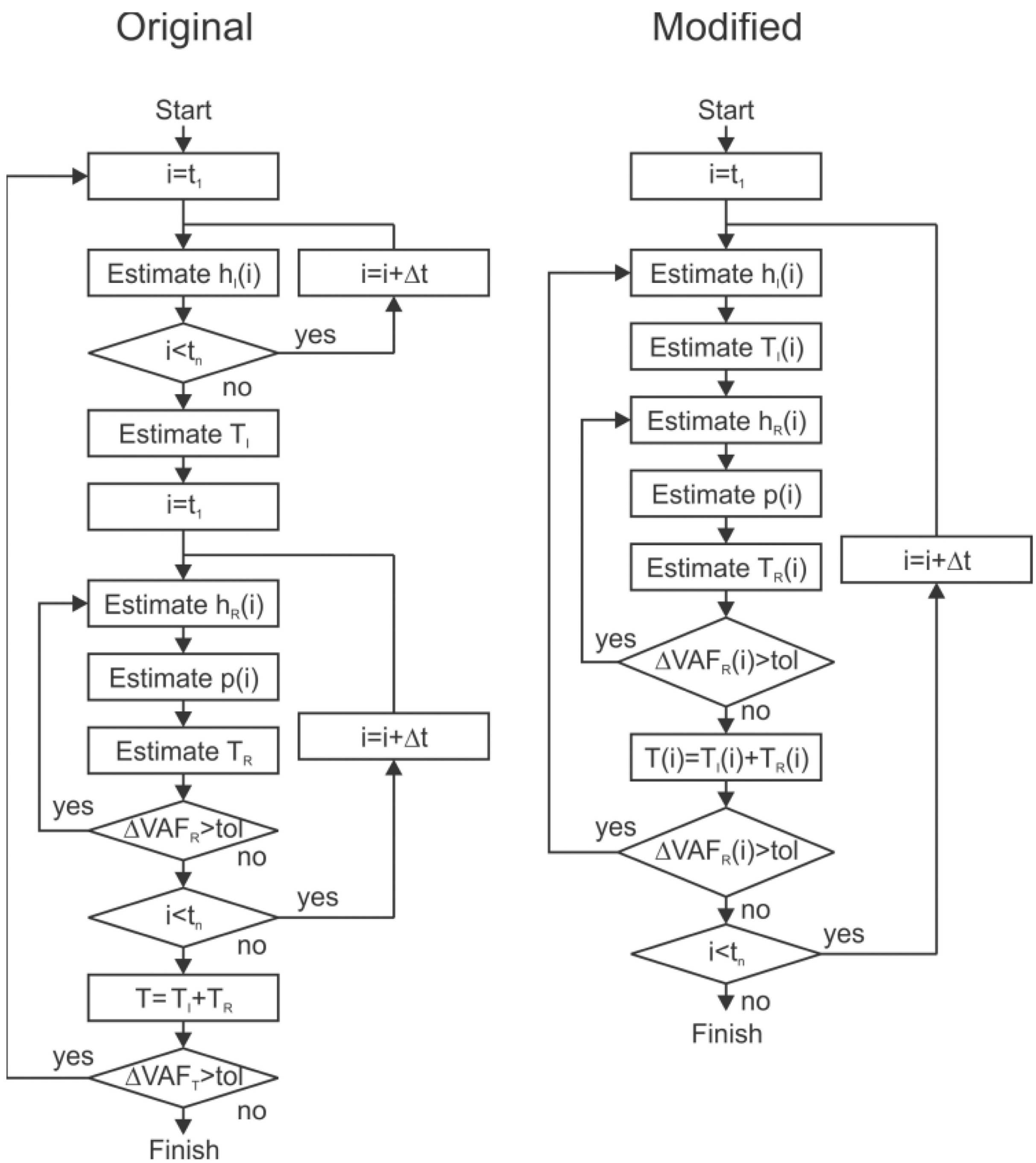


**Fig. 1.**
Flow diagram of original and modified TVPC algorithm. Original algorithm computed stiffness for all time points within each PC loop. Conversely, the modified algorithm performed PC loop for each time point.

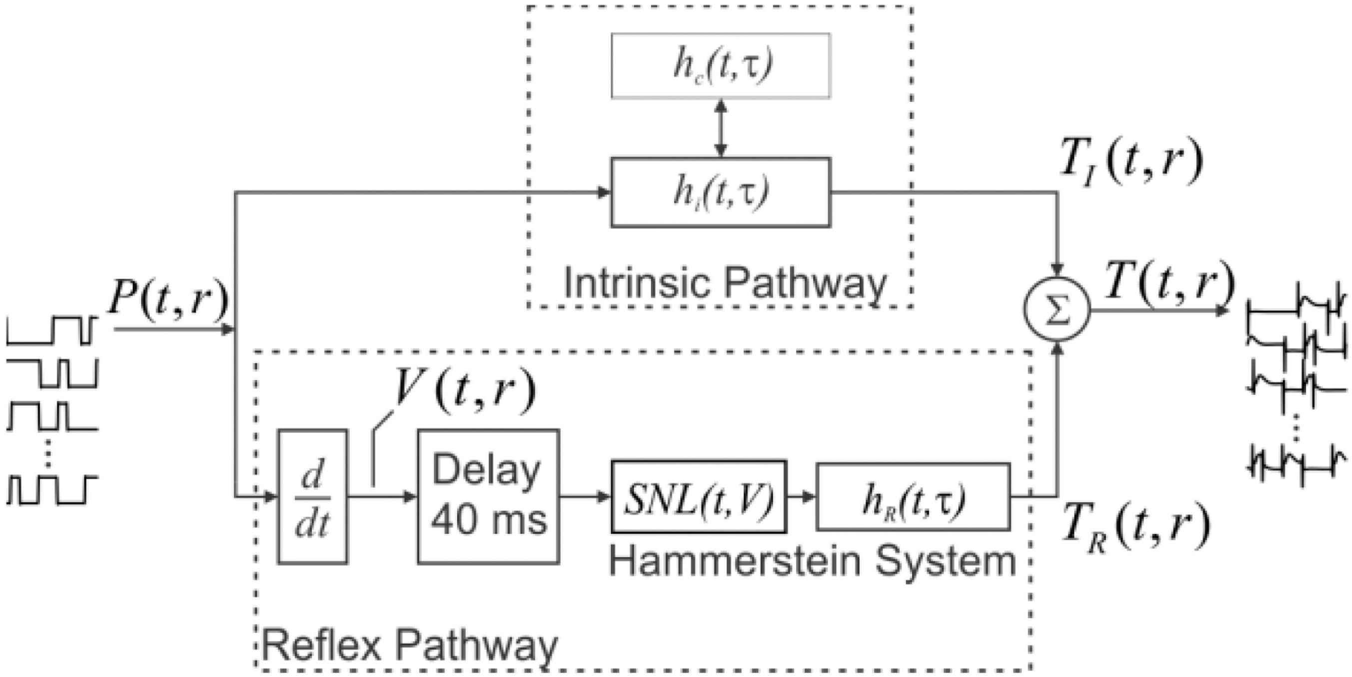


**Fig. 2.**
Schematic of the PC model of time-varying joint stiffness.

NIH-PA Author Manuscript
NIH-PA Author Manuscript



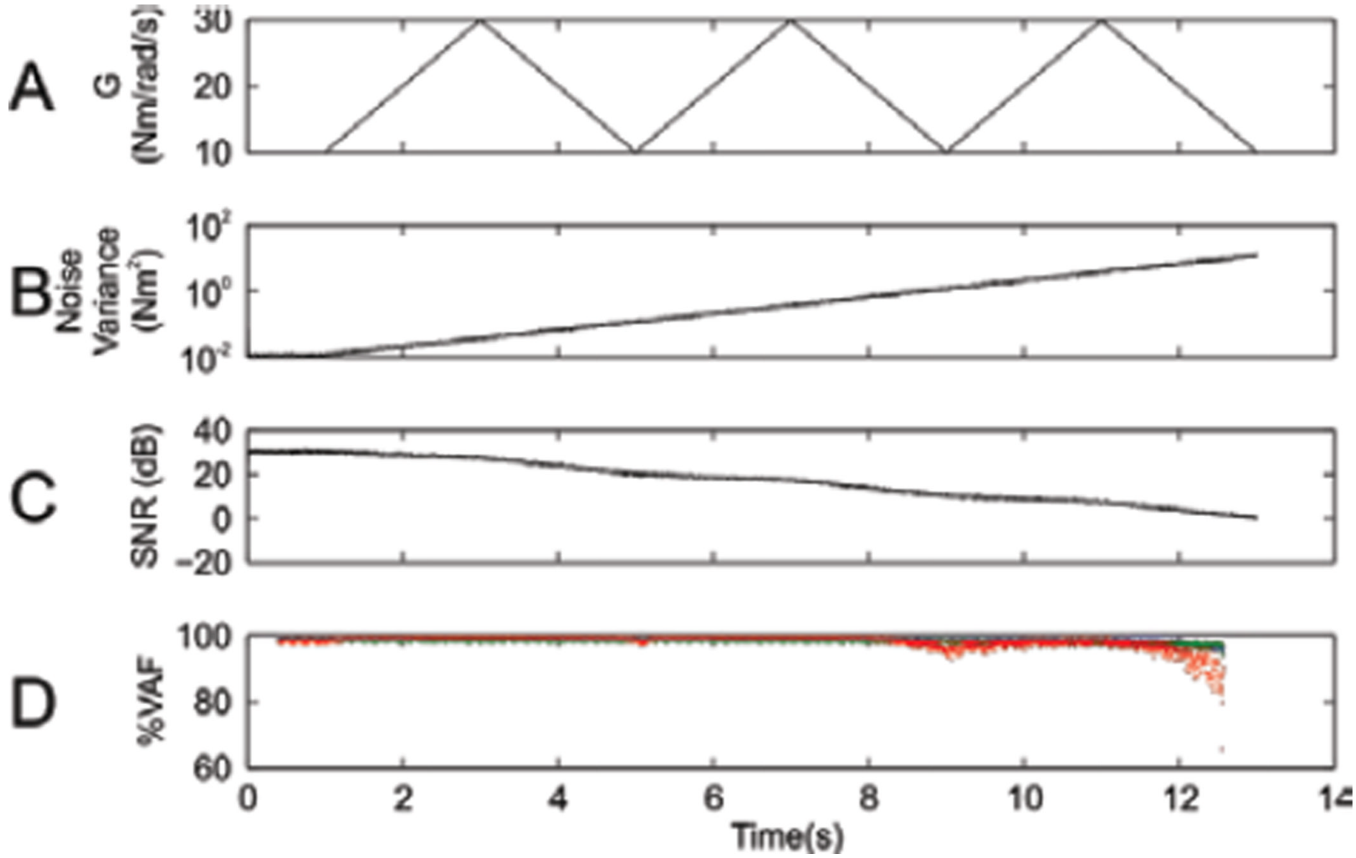


**Fig. 3.**
A) Simulated TV reflex stiffness gain (G), B) simulated output noise variance and C) resultant SNR. D) Total (blue) and intrinsic (green) torque estimates were accurate at all times, while reflex (red) torque estimates were accurate until approximately 12 s.

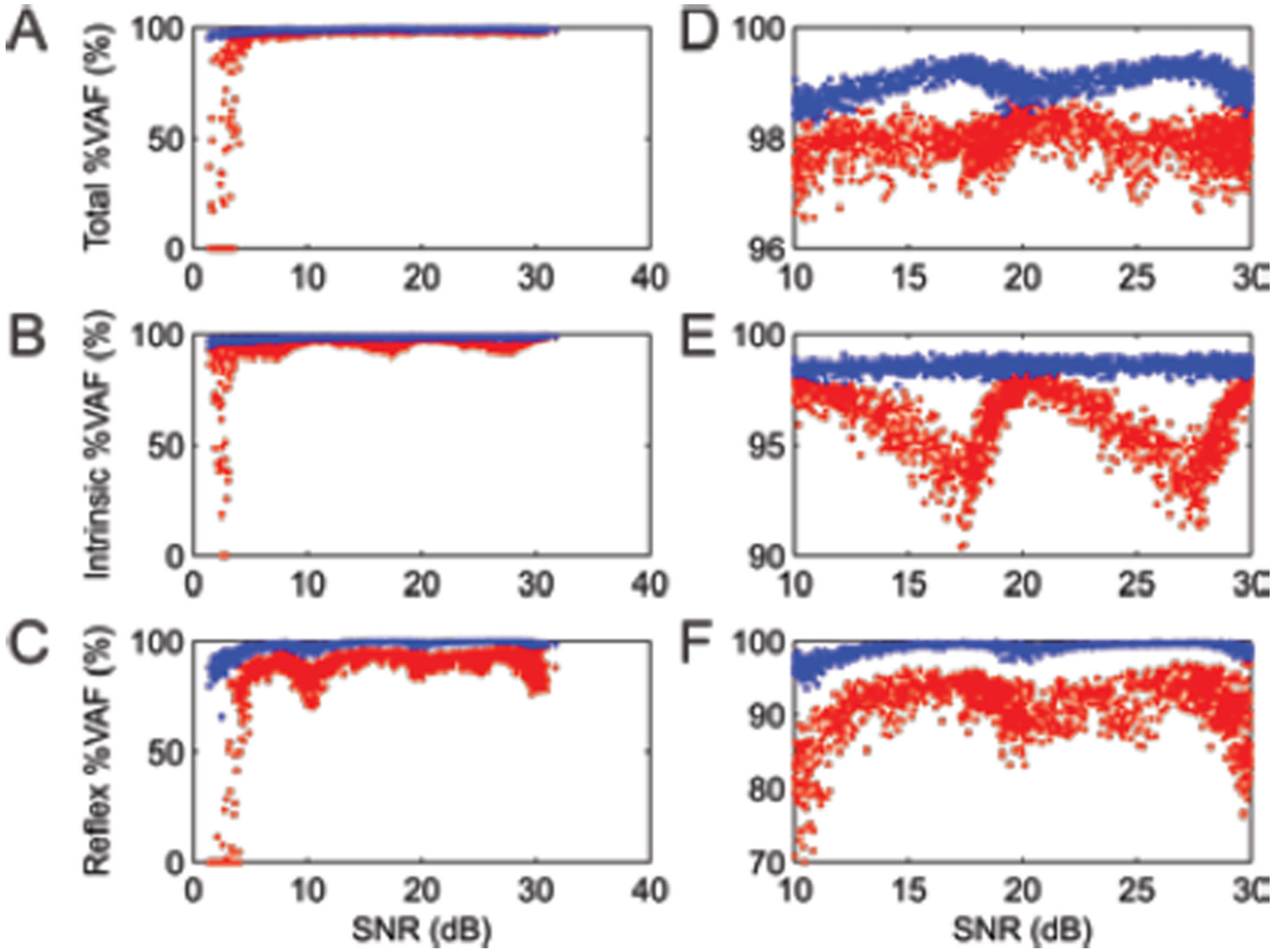


**Fig. 4.**

A–C) The modified algorithm (blue) greatly outperformed the original algorithm (red) at low SNR, and D–F) also slightly outperformed even at higher SNR levels.

**Table 1**

| | Ver. 1 | Ver. 2 | Ver. 3 |
|---|---|---|---|
| Total Time (s) | 218 | 94 | 30 |
| # PC | 590 | 590 | 204 |
| Iterations | | | |
| Time per PC iteration (s) | 0.37 | 0.16 | 0.15 |