Time-Varying Delay Estimation using Common Local All-Pass Filters with Application to Surface Electromyography

Christopher Gilliam[1], Adrian Bingham[1], Thierry Blu[2], Beth Jelfs[1]

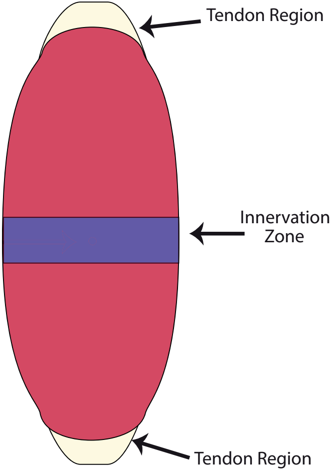
[1] School of Engineering, RMIT University, Australia
[2] Department of Electronic Engineering, The Chinese University of Hong Kong

20th April 2018

Outline

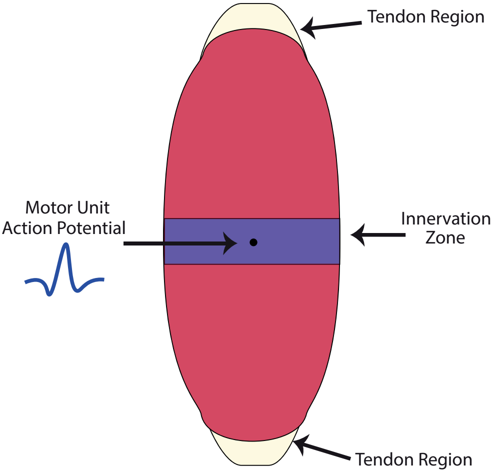
1. Introduction
   - Conduction velocity in surface electromyography (sEMG)
   - Equivalent to Time-Varying Delay Estimation
2. Estimating a Delay using All-Pass Filters
   - Shifting by a constant delay $\Longrightarrow$ All-pass filtering
   - Time-varying delay obtained from Local All-Pass (LAP) filters
   - Estimate delay common to a group of signals $\Rightarrow$ Common LAP
3. Evaluation Results
   - Synthetic sEMG data
   - Experimental data $\Longrightarrow$ High density sEMG recordings
4. Conclusions

Conduction Velocity



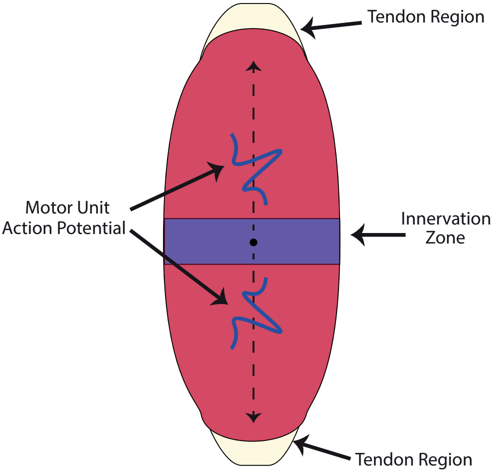
Describes the speed of propagation of motor unit action potentials (MUAPs) along the muscle

Conduction Velocity



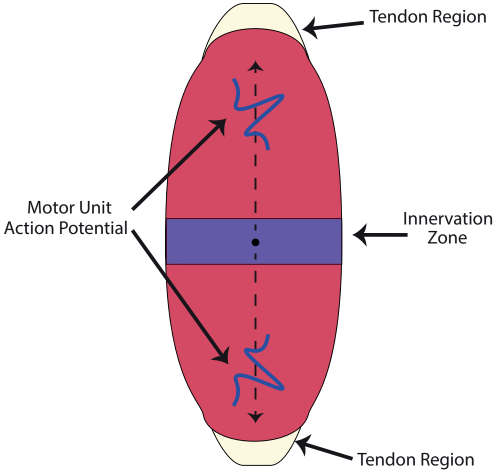
Describes the speed of propagation of motor unit action potentials (MUAPs) along the muscle

Conduction Velocity



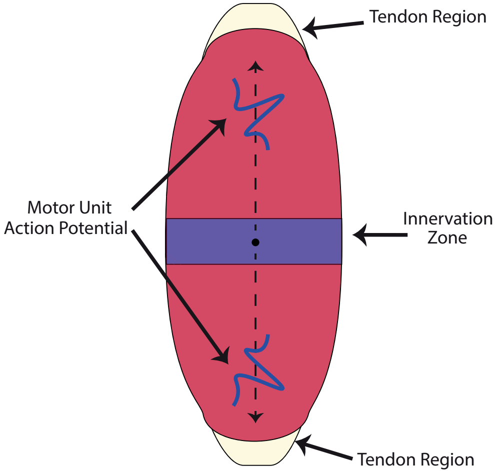
Describes the speed of propagation of motor unit action potentials (MUAPs) along the muscle

Conduction Velocity



Describes the speed of propagation of motor unit action potentials (MUAPs) along the muscle

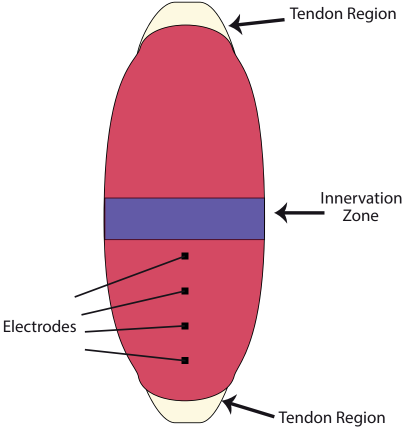
Conduction Velocity



↬ Important factor in the study of muscle pathology, fatigue or pain

Estimating Conduction Velocity from sEMG

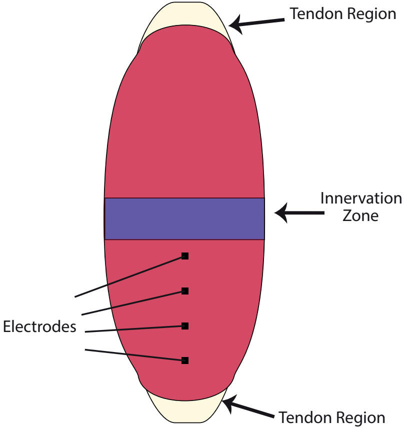
Estimate conduction velocity $\Longrightarrow$ Delay estimation



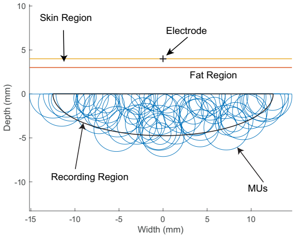
- Using high-density electrode arrays:
  - Measure the propagation of the signal

Estimating Conduction Velocity from sEMG

Estimate conduction velocity $\Longrightarrow$ Delay estimation

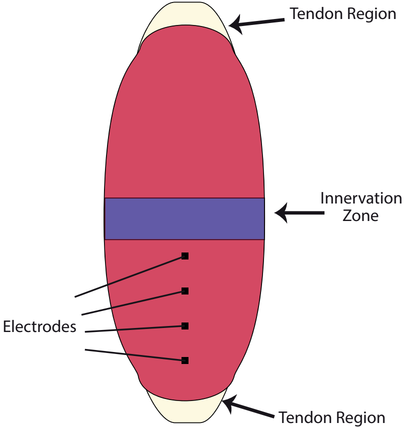


- Using high-density electrode arrays:
  - Measure the propagation of the signal
- Difficulties:
  - sEMG contains many MUAPs

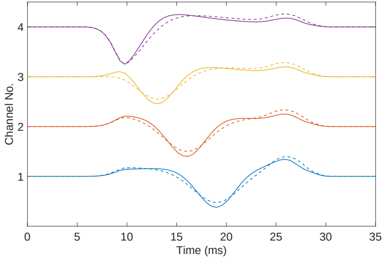


Estimating Conduction Velocity from sEMG

Estimate conduction velocity $\implies$ Delay estimation

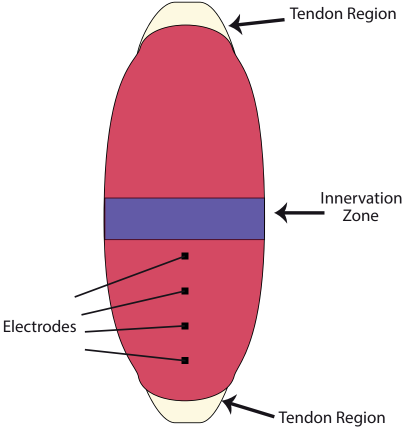


- Using high-density electrode arrays:
  - Measure the propagation of the signal
- Difficulties:
  - Non-constant MUAP shape

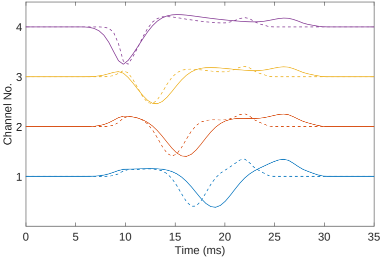


Estimating Conduction Velocity from sEMG

Estimate conduction velocity $\Longrightarrow$ Delay estimation

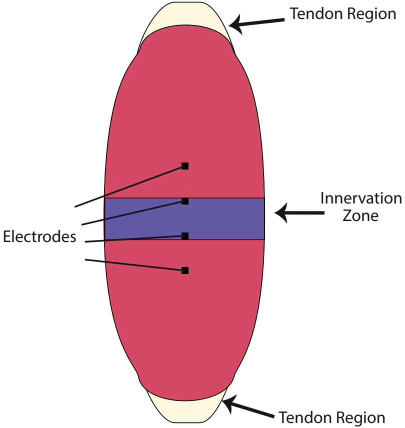


- Using high-density electrode arrays:
  - Measure the propagation of the signal
- Difficulties:
  - Non-constant conduction velocity

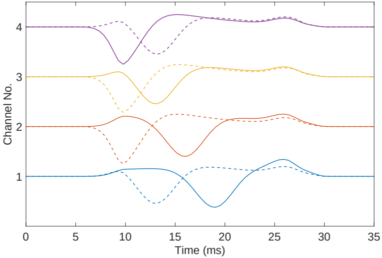


Estimating Conduction Velocity from sEMG

Estimate conduction velocity $\implies$ Delay estimation

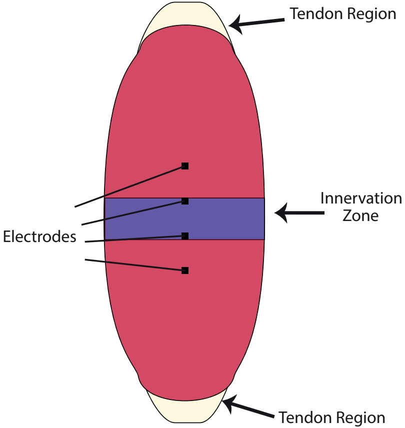


- Using high-density electrode arrays:
  - Measure the propagation of the signal
- Difficulties:
  - Array placement

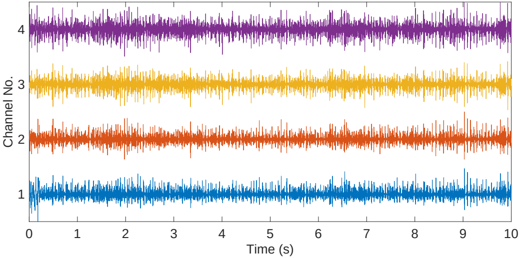


Estimating Conduction Velocity from sEMG

Estimate conduction velocity $\Longrightarrow$ Delay estimation

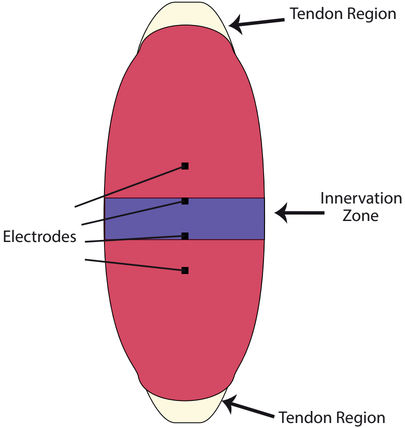


- Using high-density electrode arrays:
  - Measure the propagation of the signal
- Difficulties:
  - EMG signals are noisy

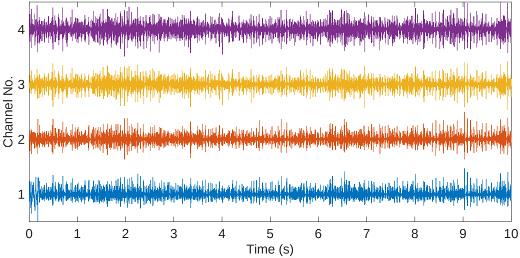


Estimating Conduction Velocity from sEMG

Estimate conduction velocity $\implies$ Delay estimation



- Using high-density electrode arrays:
  - Measure the propagation of the signal
- Difficulties:
  - EMG signals are noisy



↬ Time-varying delay estimation with unknown waveforms

Time-Varying Delay Estimation

The problem:

Multi-channel recordings:

$$\begin{cases} g_1(t) = f(t) + e_1(t) \\ g_2(t) = f\big(t - \tau(t)\big) + e_2(t) \\ g_3(t) = f\big(t - 2\,\tau(t)\big) + e_3(t) \\ \qquad\vdots \\ g_N(t) = f\big(t - (N-1)\tau(t)\big) + e_N(t) \end{cases}$$

where

- $g_n(t)$ is the signal from the nth electrode
- $f(t)$ is the signal of interest
- $\tau(t)$ is the time-varying delay
- $e_n(t)$ is Gaussian noise

Time-Varying Delay Estimation

The problem:

Multi-channel recordings:

$$\begin{cases} g_1(t) = f(t) + e_1(t) \\ g_2(t) = f\big(t - \tau(t)\big) + e_2(t) \\ g_3(t) = f\big(t - 2\,\tau(t)\big) + e_3(t) \\ \qquad\vdots \\ g_N(t) = f\big(t - (N-1)\tau(t)\big) + e_N(t) \end{cases}$$

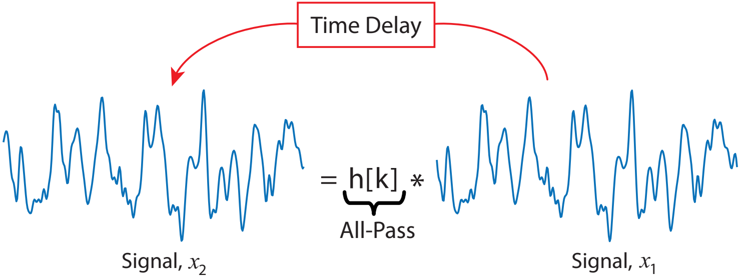
Our Approach:

Common Local All-Pass Filter algorithm:

- ↬ Robust and very accurate
- ↬ Automatically identify Innervation Zone
- ↬ Uses all of the electrode signals

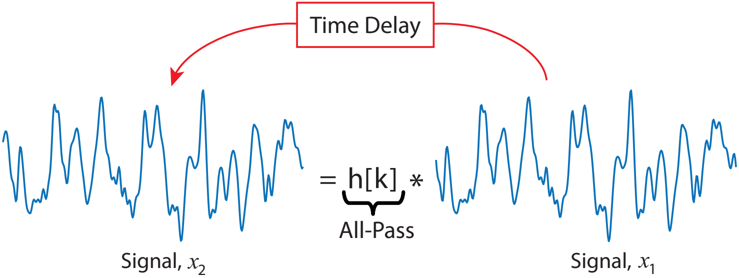
All-Pass Filter Framework - Concept 1

Constant delay $\tau \implies$ Filtering Signal 1 with All-Pass Filter h



All-Pass Filter Framework - Concept 1

Constant delay $\tau \implies$ Filtering Signal 1 with All-Pass Filter h

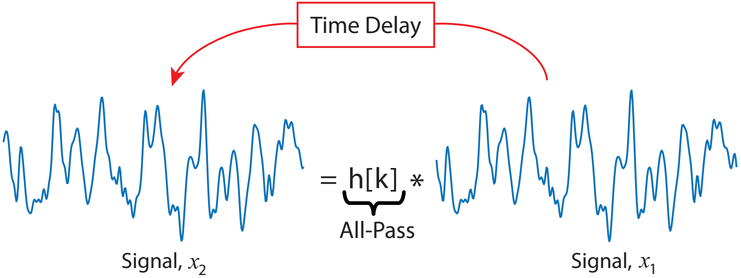


Shifting in Frequency:

$$X_2(\omega) = \underbrace{X_1(\omega)\, \mathrm{e}^{-j\tau\omega}}_{=\ \text{Filtering Operation}} \xrightarrow{\text{Define Filter}} \underbrace{H(\omega) = \mathrm{e}^{-j\tau\omega}}_{=\ \text{All-Pass}}$$

All-Pass Filter Framework - Concept 1

Constant delay $\tau \implies$ Filtering Signal 1 with All-Pass Filter h



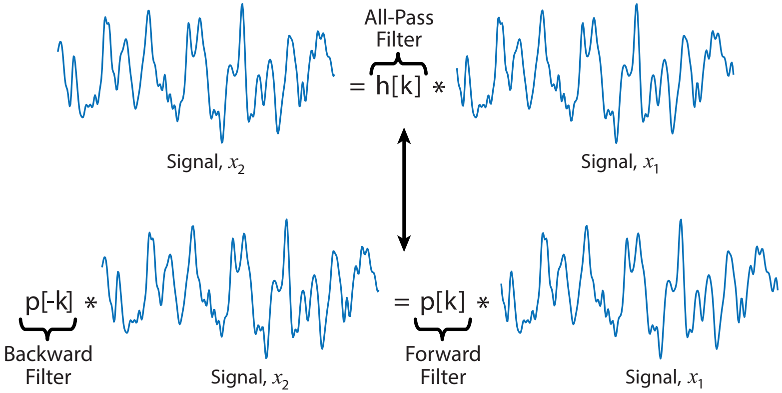
↬ Estimation of All-Pass Filter = Estimation of time delay
↬ No assumption on the size of the delay

All-Pass Filter Framework - Concept 2

Any all-pass filter has a rational structure $\implies H(\omega) = \dfrac{P(\omega)}{P(-\omega)}$

All-Pass Filter Framework - Concept 2

Any all-pass filter has a rational structure $\implies H(\omega) = \dfrac{P(\omega)}{P(-\omega)}$



↬ Linearise All-Pass Filtering

All-Pass Filter Framework - Concept 3

Approximate p with a few known real filters

$$p_{\text{app}}[k] = \sum_{l=0}^{L-1} c_l p_l[k] \quad \Longrightarrow \quad \text{Estimate coefficients } c_l$$

All-Pass Filter Framework - Concept 3

Approximate p with a few known real filters

$$p_{\text{app}}[k] = \sum_{l=0}^{L-1} c_l p_l[k] \quad \Longrightarrow \quad \text{Estimate coefficients } c_l$$

A good choice of filters $\Longrightarrow$ Span the derivatives of an isotropic function

*T. Blu, P. Moulin & C. Gilliam, "*Approximation order of the LAP optical flow algorithm*", Proc IEEE Int. Conf. Image Processing, Québec city, Canada, September 27–30 2015.

All-Pass Filter Framework - Concept 3

Approximate p with a few known real filters

$$p_{\text{app}}[k] = \sum_{l=0}^{L-1} c_l p_l[k] \quad \Longrightarrow \quad \text{Estimate coefficients} \quad c_l$$

A good choice of filters $\Longrightarrow$ Span the derivatives of an isotropic function

$$p_0[k] = \exp\left(-\frac{k^2}{2\sigma^2}\right) \quad \text{where} \quad \sigma = \frac{R}{2} - 0.2$$

$$\text{First derivatives} \quad \Longrightarrow \quad p_1[k] = k\, p_0[k]$$

↬ Filters are scalable $\Longrightarrow$ Estimate both large and small delays

↬ Linked to R, the half support of the filters

*T. Blu, P. Moulin & C. Gilliam, "*Approximation order of the LAP optical flow algorithm*", Proc IEEE Int. Conf. Image Processing, Québec city, Canada, September 27–30 2015.

All-Pass Filter Framework - Concept 3

Approximate p with a few known real filters

$$p_{\text{app}}[k] = \sum_{l=0}^{L-1} c_l p_l[k] \quad \Longrightarrow \quad \text{Estimate coefficients } c_l$$

Finally....

$$\text{Set } c_0 = 1 \quad \Longrightarrow \quad \text{Need to estimate 1 coefficients}$$

$\looparrowright$ Solve using Least Mean Squares $\Longrightarrow$ Linear system of equations

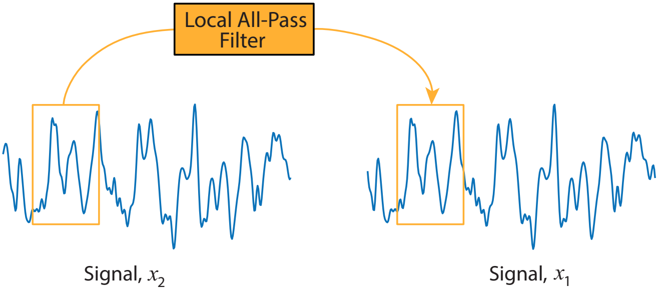
$$\min_{\{c_l\}} \sum_{k \in \mathbb{Z}} \left| p_{\text{app}}[k] * x_1[k] - p_{\text{app}}[-k] * x_2[k] \right|^2$$

$\looparrowright$ Extract estimate of delay from All-Pass Filter

Local All-Pass (LAP) Algorithm

Central Assumption:

Assume delay is constant within a local region ⇒ Local All-Pass Filters



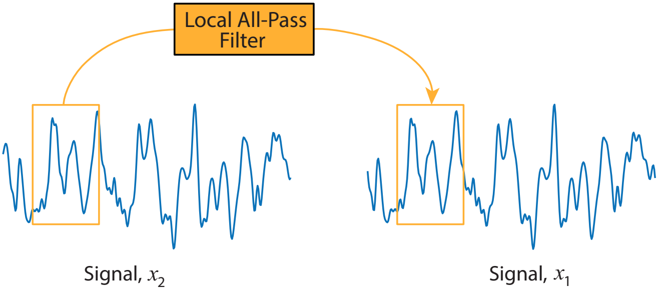
At central sample ⟹ Estimate local all-pass filter

↬ Extract delay from the estimate of the filter

Local All-Pass (LAP) Algorithm

Central Assumption:

Assume delay is constant within a local region ⇒ Local All-Pass Filters



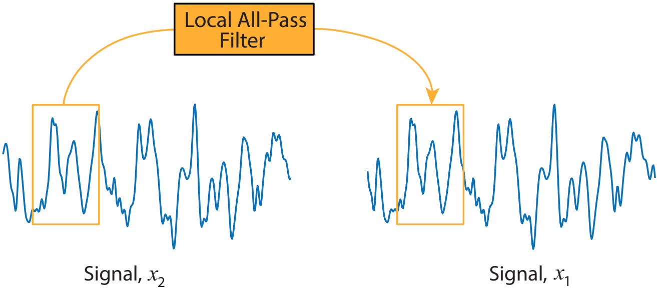
Move region (one sample change) $\implies$ Estimate new all-pass filter

Very efficient to solve $\implies$ Convolutions and fixed-point multiplication

Local All-Pass (LAP) Algorithm

Central Assumption:

Assume delay is constant within a local region $\Rightarrow$ Local All-Pass Filters



Move region (one sample change) $\implies$ Estimate new all-pass filter

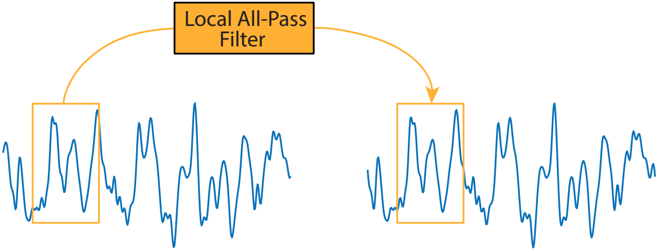
Very efficient to solve $\implies$ Convolutions and fixed-point multiplication

$\looparrowright$ Size of the region $= R$, half support of the filters

Local All-Pass (LAP) Algorithm

Central Assumption:

Assume delay is constant within a local region ⇒ Local All-Pass Filters



Signal, x_2 Signal, x_1

End Result:

↬ Per sample estimate of the time-varying delay

Estimating a Common Time-Varying Delay

The ensemble of signals:

$$\left.\begin{aligned} x_1(t) &= f(t) \\ x_2(t) &= f\big(t - \tau(t)\big) \\ x_3(t) &= f\big(t - 2\tau(t)\big) \\ &\vdots \\ x_N(t) &= f\big(t - (N-1)\tau(t)\big) \end{aligned}\right\}$$

Characterised by time varying delay

Estimating a Common Time-Varying Delay

The ensemble of signals:

$$\left.\begin{aligned} x_1(t) &= f(t) \\ x_2(t) &= f\big(t-\tau(t)\big) \\ x_3(t) &= f\big(t-2\tau(t)\big) \\ &\vdots \\ x_N(t) &= f\big(t-(N-1)\tau(t)\big) \end{aligned}\right\} \Longrightarrow \left\{\begin{aligned} x_2(t) &= x_1\big(t-\tau(t)\big) \\ x_3(t) &= x_2\big(t-\tau(t)\big) \\ &\vdots \\ x_N(t) &= x_{N-1}\big(t-\tau(t)\big) \end{aligned}\right.$$

Estimating a Common Time-Varying Delay

The ensemble of signals:

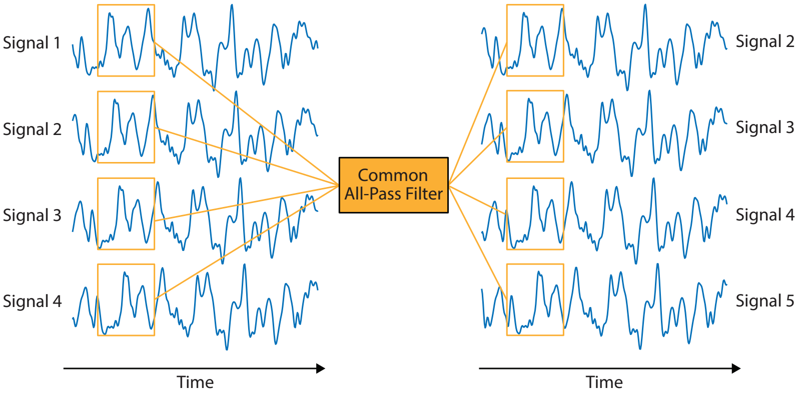
$$\left.\begin{aligned} x_1(t) &= f(t) \\ x_2(t) &= f\big(t-\tau(t)\big) \\ x_3(t) &= f\big(t-2\tau(t)\big) \\ &\;\vdots \\ x_N(t) &= f\big(t-(N-1)\tau(t)\big) \end{aligned}\right\} \Longrightarrow \left\{\begin{aligned} x_2(t) &= x_1\big(t-\tau(t)\big) \\ x_3(t) &= x_2\big(t-\tau(t)\big) \\ &\;\vdots \\ x_N(t) &= x_{N-1}\big(t-\tau(t)\big) \end{aligned}\right.$$

Key Observation:

Same time-varying delay $\tau(t)$ between each pair of signals
$\looparrowright$ Adapt LAP to multiple signals

Common Local All-Pass Filter

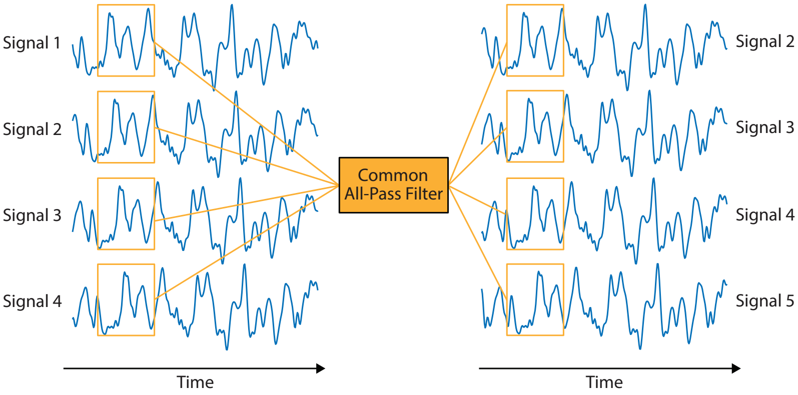
Our approach:



Within local regions $\Longrightarrow$ Estimate a common All-Pass Filter

Common Local All-Pass Filter

Our approach:

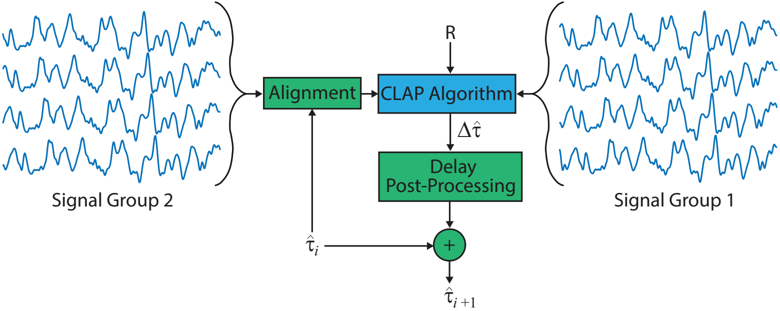


Common Local All-Pass (CLAP) Algorithm

Estimate a single time-varying delay common across a group of signals

Multi-Scale Framework

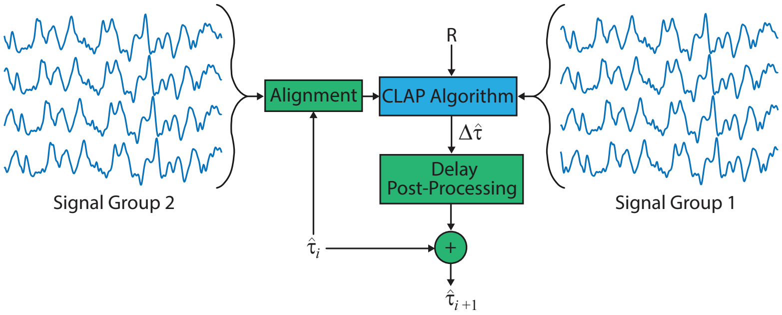
Iterative framework:



Estimate faster variations in the delay $\Longrightarrow$ Change R (size of filters)

Multi-Scale Framework

Iterative framework:



Alignment $\Rightarrow$ Warp signals using delay estimate

Post-Processing:
- Inpainting $\Longrightarrow$ Remove erroneous delay estimates
- Smooth estimate using Gaussian filtering

Application to sEMG

Step 1:

sEMG signals likely to suffer from a common source of corruption across all channels

Use single differential of signals: $x_n(t) = g_{n+1}(t) - g_n(t)$

Application to sEMG

Step 2:

Automatic identification of the Innervation Zone

Innervation Zone

Point where motor neurons innervate the muscle fibres
$\looparrowright$ MUAPs propagate out from zone to tendons

Application to sEMG

Step 2:

Automatic identification of the Innervation Zone

Innervation Zone

Point where motor neurons innervate the muscle fibres
↬ MUAPs propagate out from zone to tendons

$$
\begin{aligned}
x_{n-2} &= x_{n-1}(t + \tau(t)) \\
x_{n-1} &= x_n(t + \tau(t)) \\
x_n &\longrightarrow \text{Innervation Zone} \\
x_{n+1} &= x_n(t - \tau(t)) \\
x_{n+2} &= x_{n+1}(t - \tau(t))
\end{aligned}
$$

Application to sEMG

Step 2:

Automatic identification of the Innervation Zone

Innervation Zone

Point where motor neurons innervate the muscle fibres
$\looparrowright$ MUAPs propagate out from zone to tendons

Solution:

- Run LAP pair-wise on adjacent signals & calculate mean delay
- Find point where the sign of the delay changes
- Reverse order of processing for signals above the zone

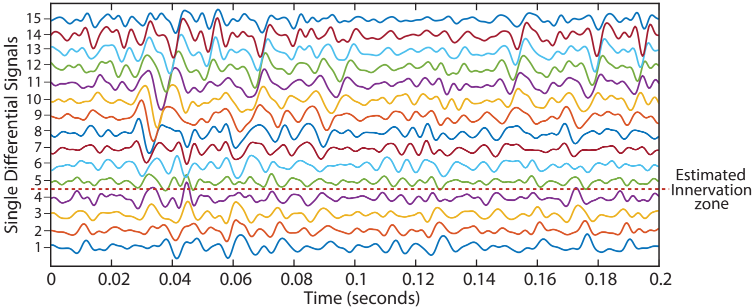
Application to sEMG

Step 2:

Automatic identification of the Innervation Zone

Innervation Zone

Point where motor neurons innervate the muscle fibres
↬ MUAPs propagate out from zone to tendons



Evaluation on Synthetic sEMG Data

Synthetic Data Model[1]:

[1] P. Ravier *el al.*, '*Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface eletromyogram*', Biomed. Signal Process. Control, 2015.
[2] E. Shwedyk *el al.*, '*A nonstationary model for the electromyogram*', IEEE Trans. Biomed. Eng., 1977.

Evaluation on Synthetic sEMG Data

Synthetic Data Model[1]:

1st Channel:
White Gaussian noise filtered using FIR filter with EMG-like spectral properties[2].

[1] P. Ravier *el al.*, '*Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface eletromyogram*', Biomed. Signal Process. Control, 2015.

[2] E. Shwedyk *el al.*, '*A nonstationary model for the electromyogram*', IEEE Trans. Biomed. Eng., 1977.

Evaluation on Synthetic sEMG Data

Synthetic Data Model[1]:

1st Channel:
White Gaussian noise filtered using FIR filter with EMG-like spectral properties[2].

$\Rightarrow$

Generate other channels via interpolation using $\tau(t)$

[1] P. Ravier *el al.*, '*Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface eletromyogram*', Biomed. Signal Process. Control, 2015.
[2] E. Shwedyk *el al.*, '*A nonstationary model for the electromyogram*', IEEE Trans. Biomed. Eng., 1977.

Evaluation on Synthetic sEMG Data

Synthetic Data Model[1]:

1st Channel: White Gaussian noise filtered using FIR filter with EMG-like spectral properties[2].

$\Rightarrow$

Generate other channels via interpolation using $\tau(t)$

$\Rightarrow$

Each channel corrupted by Gaussian noise (shaped by FIR filter to simulate sEMG acquisition device)

[1] P. Ravier *el al.*, '*Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface eletromyogram*', Biomed. Signal Process. Control, 2015.

[2] E. Shwedyk *el al.*, '*A nonstationary model for the electromyogram*', IEEE Trans. Biomed. Eng., 1977.

Evaluation on Synthetic sEMG Data

Synthetic Data Model[1]:

1st Channel: White Gaussian noise filtered using FIR filter with EMG-like spectral properties[2].	$\Rightarrow$	Generate other channels via interpolation using $\tau(t)$	$\Rightarrow$	Each channel corrupted by Gaussian noise (shaped by FIR filter to simulate sEMG acquisition device)

Model of Conductance Velocity:

$$\mathsf{CV}(t) = 4 + 2\sin(2\pi\, 0.2t/F_s)$$

where $F_s = 2048\,\mathsf{Hz}$ is the sampling frequency

Biologically plausible $\implies$ Velocities range between $2\,\mathsf{m/s}$ to $6\,\mathsf{m/s}$

[1] P. Ravier *el al.*, '*Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface eletromyogram*', Biomed. Signal Process. Control, 2015.
[2] E. Shwedyk *el al.*, '*A nonstationary model for the electromyogram*', IEEE Trans. Biomed. Eng., 1977.

Evaluation on Synthetic sEMG Data

Synthetic Data Model[1]:

1st Channel: White Gaussian noise filtered using FIR filter with EMG-like spectral properties[2].

$\Rightarrow$

Generate other channels via interpolation using $\tau(t)$

$\Rightarrow$

Each channel corrupted by Gaussian noise (shaped by FIR filter to simulate sEMG acquisition device)

Model of Conductance Velocity:

$$\text{CV}(t) = 4 + 2\sin(2\pi\, 0.2t/F_s)$$

where $F_s = 2048\,\text{Hz}$ is the sampling frequency

Biologically plausible $\implies$ Velocities range between $2\,\text{m/s}$ to $6\,\text{m/s}$

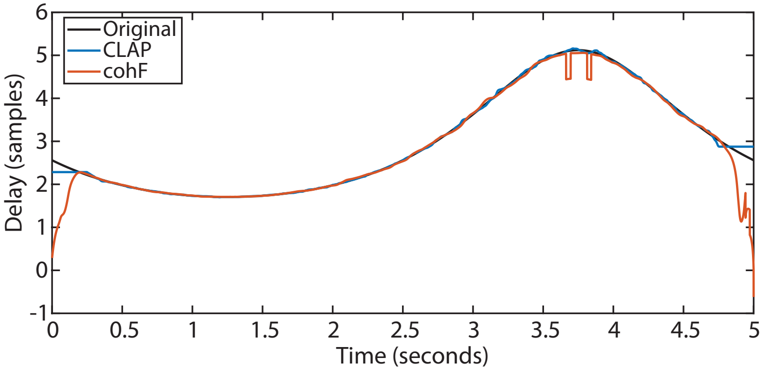
$$\looparrowright \quad \tau(t) = F_s \frac{\Delta e}{\text{CV}(t)}$$

where $\Delta e = 5\,\text{mm}$ is the inter electrode distance

[1] P. Ravier *el al.*, '*Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface eletromyogram*', Biomed. Signal Process. Control, 2015.
[2] E. Shwedyk *el al.*, '*A nonstationary model for the electromyogram*', IEEE Trans. Biomed. Eng., 1977.

Evaluation on Synthetic sEMG Data - Results

Noiseless Data:

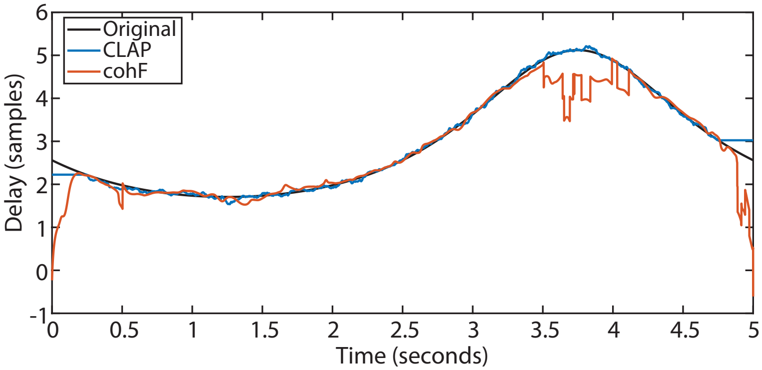


Estimation using 6 signals and window size of $0.25\,\text{s}$ (512 samples)

cohF = Fourier Phase Coherency (same window size)

Evaluation on Synthetic sEMG Data - Results

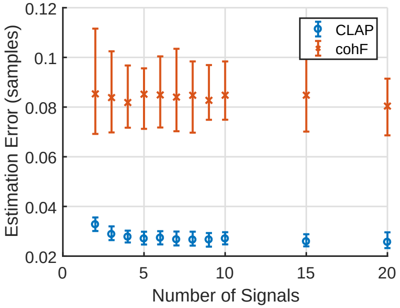
SNR $= 10\,$dB:



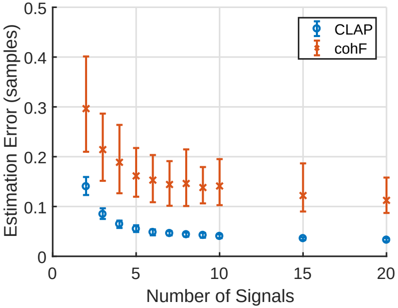
Estimation using 6 signals and window size of $0.25\,$s (512 samples)

cohF = Fourier Phase Coherency (same window size)

Synthetic sEMG Data - Varying Number of Signals



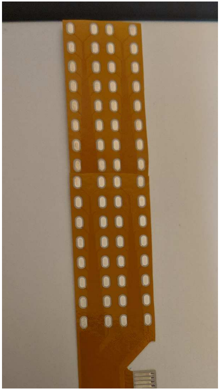
SNR $= 30\,$dB



SNR $= 10\,$dB

Error bars $= 5^{\text{th}}$ and 95^{th} quantiles
Values averaged over 100 realisations of the data

Evaluation on Experimental HD-sEMG Data



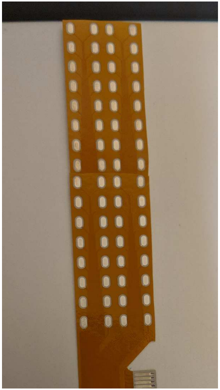
Electrode Array

Example of set up

4 $\times$ 16 HD-sEMG array placed on biceps brachii (parallel to muscle fibres)

Participants pull on fixed cable $\Rightarrow$ 40% & 80% max voluntary contractions

Evaluation on Experimental HD-sEMG Data



Electrode Array

Example of set up

Validate CLAP algorithm performance using surrogate data

Surrogates generated by iterative amplitude adjusted FFT

Evaluation on Experimental HD-sEMG Data - Results

Subject	MVC	Surrogates		Data	
		Avg $\bar{\tau}$	Var $\bar{\tau}$	$\bar{\tau}$	CV (m/s)
1	40%	-0.001	0.000	2.503	4.89
	80%	-0.003	0.000	2.363	5.20
2	40%	-0.001	0.000	2.320	5.28
	80%	0.000	0.002	2.399	5.11
3	40%	0.001	0.000	1.690	7.26
	80%	-0.002	0.001	1.726	7.13

$\bar{\tau}$ = Time averaged delay and MVC = Maximum Voluntary Contraction

100 surrogates per dataset

Evaluation on Experimental HD-sEMG Data - Results

		Surrogates		Data		
Subject	MVC	Avg $\bar{\tau}$	Var $\bar{\tau}$	$\bar{\tau}$	CV (m/s)	Change CV (m/s^2)
1	40%	-0.001	0.000	2.503	4.89	-0.001
	80%	-0.003	0.000	2.363	5.20	-0.029
2	40%	-0.001	0.000	2.320	5.28	-0.005
	80%	0.000	0.002	2.399	5.11	-0.017
3	40%	0.001	0.000	1.690	7.26	-0.001
	80%	-0.002	0.001	1.726	7.13	-0.017

$\bar{\tau}$ = Time averaged delay and MVC = Maximum Voluntary Contraction

100 surrogates per dataset

Conclusions

- Muscle conduction velocity estimation
  - Equivalent to time-varying delay estimation
  - Using HD-sEMG $\Longrightarrow$ Delay estimation across multiple channels
- Framework for estimating a delay using all-pass filters
  - Delay estimate $\Longrightarrow$ Local All-Pass Filters
  - Multiple channels $\Longrightarrow$ Common Local All-Pass Filters
  - Estimate a single time-varying delay common across a group of signals
- Demonstration of the CLAP algorithm
  - Able to automatically estimate the Innervation Zone
  - Synthetic data $\Longrightarrow$ Robust and accurate
  - Experimental data $\Longrightarrow$ Biologically plausible CV values & validated via surrogate testing

The End

Thank you for listening