Brain tissue temperature dynamics during functional activity and possibilities for Imaging

by

GREGGORY H. ROTHMEIER

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Masters of Science

in the College of Arts and Sciences

Georgia State University

2012

Brain tissue temperature dynamics during functional activity and possibilities for Imaging

by

GREGGORY H. ROTHMEIER

Committee Chair:	A. G. Unil Perera
Committee:	Mukesh Dhamala
	Brian Thoms
	D. Michael Crenshaw

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2012

Dedication

Mama.

Acknowledgements

Contents

List of Tables

List of Figures

Brain tissue temperature dynamics during functional activity and possibilities for

Imaging

A thesis

presented in Partial Fulfilment of Requirements for the Degree of Master of Science in the

College of Arts and Sciences

Georgia State University

2012

by

Greggory Rothmeier

Committee:

A. G. Unil Perera, Chair

Mukesh Dhamala, Member

Brian Thoms, Member

D. Michael Crenshaw, Member

April 1, 2012

Date

Dick Miller
Department Chair

Chapter 1

Introduction

Chapter 2

Calculating Temperature Changes using the fMRI BOLD Response

2.1 Background

2.1.1 Generation of the Blood Oxygen Level Dependent (BOLD) Response

Since its invention in the 1950's [1] and later development in the 1970's [2], Magnetic Resonance Imaging (MRI) has allowed physicians and scientists a detailed view within the human body.

2.1.2 Previously Proposed Temperature Models

Current efforts to model temperature changes be can categorized into two classes. The first class approaches the problem by considering a single voxel deep within the brain (single-voxel approach) while the second approach considers the brain and head as an entire system (multi-voxel approach). Each of these methods has their own pros and cons which will be discussed below.

Single-Voxel Approach

A single-voxel model of temperature was first proposed by SOMEONE, but has been refined over the past HOWLONG years CITEABUNCH to include more terms. Although different approaches consider different contributions to the temperature change, they all narrow the problem down to a single voxel which is usually 2mm x 2mm x 2mm. By simplifying the model, the heat equation can be simplified and the calculation is much easier to undertake. However, since the brain is not homogenous, the values used for parameters
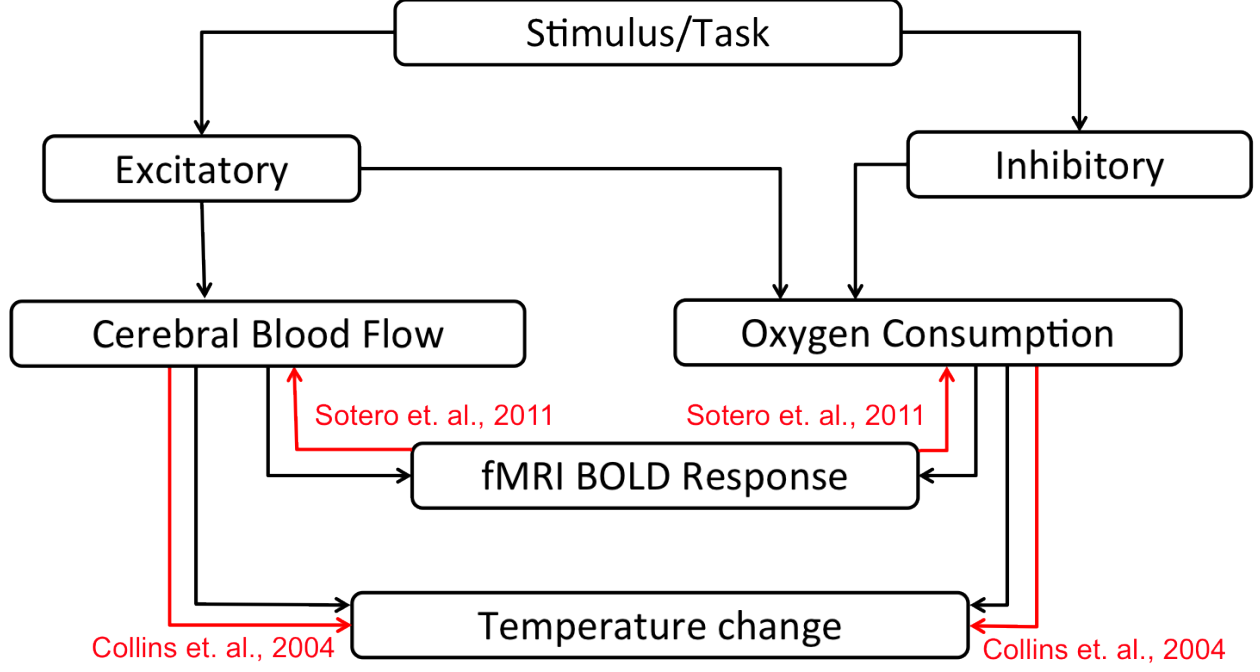
Figure 2.1: Generation of the fMRI BOLD response from changes in neuronal activity. Black arrows indicate a causal relationship while red arrows indicate existing models for the relationship. Modified from Sotero and Trujillo-Barreto [3]

such as heat production and thermal conductivity are taken from an average of the tissues. As a result, this reduces the possible accuracy of such a model when applied to a subject. The most recently published iteration of a single-voxel model was published by Sotero and Iturria-Medina [4]. The basis of this model is a modification of the Penne's Bioheat Equation [5, 4].

$$C_t \frac{dT(t)}{dt} = (\Delta H^\circ - \Delta H_b) CMRO_2 \mid_0 m(t) - \rho_b C_b CBF \mid_0 f(t)(T(t) - T_a) - \frac{C_t}{\tau}(T(t) - T_0) \qquad (2.1)$$

where BLA BLA BLA. One advantage of using eq. (2.1) is that the resting state temperature can be analytically determined by substituting $\frac{dT(t)}{dt} = 0$ [4].

$$T_0 = T_a + \frac{(\Delta H \mid^\circ - \Delta H_b) CMRO_2 \mid_0}{\rho_B C_B CBF \mid_0} \qquad (2.2)$$

If the values provided in table 2.1 are substitued into eq. (2.2), a resting temperature of 37.3057°C is found. Since the resting temperature is always greater than the arterial blood temperature, it limits the ability of the model to account for all experimental results.

3

While eq. (2.1) is appears complicated, conceptually the equation can be easily understood.

$$change\ in\ temperature\ =\ heat\ generated\ by\ metabolism - heat\ lost\ to\ convection - heat\ lost\ to\ conduction$$

(2.3)

The system is a balance between heat generation (metabolism) and heat transfer (conduction and convection). The direction of heat transfer by convection is determined by the difference between the voxel temperature and the arterial blood temperature $(T(t) - T_a)$. Similarly, the direction of heat transfer by conduction is determined by the difference between the voxel temperature and the temperature of the surrounding tissue $(T(t) - T_0)$. Since T_a is less than $T(0)$, an increase in blood flow $(f(t))$ will remove heat from the voxel thereby decreasing the temperature. Conversely, an increase in metabolism $(m(t))$ without a corresponding change in blood flow, will result in tissue warming.

$$f(t) = \frac{\alpha + \beta c}{b\beta} W(y(t))$$

(2.4)

$$m(t) = af^{c+1}(t)e^{-bf(t)}$$

(2.5)

$$y(t) = -\frac{b\beta}{\alpha + \beta c} \left[\frac{(A - \frac{S(t)}{S_0} - 1)}{Aa^{\beta}} \right]^{\left(\frac{1}{\alpha + \beta c}\right)}$$

(2.6)

Table 2.1: Parameters used to solve the single-voxel Penne's Bioheat Equation. (modified from Sotero and Iturria-Medina [4])

Parameter	Meaning	Value
T_a	Arterial blood temperature	$37°C$
C_{tissue}	Tissue Heat Capacity	3.664 J/(gK)
$\Delta H°$	Enthalpy released by oxidation of glucose	$4.7 10^5$ J
ΔH_b	Enthalpy used to release O_2 from hemoglobin	$2.8 10^4$ J
$CMRO_2 \mid_0$	Cerebral metabolic rate of O_2 consumption at rest	$0.0263 10^{-6}$ mol/(gs)
$CBF \mid_0$	Cerebral blood flow at rest	0.0093 cm^3/(gs)
ρ_b	Blood density	1.05 g/cm^3
C_B	Blood heat capacity	3.894 J/(gK)
τ	Time constant for conductive heat loss from the ROI to the surrounding tissue	190.52 s
a, b, c	Parameters of the gamma function fitted from E(f) vs. f	0.4492, 0.2216, -0.9872
A	Maximum BOLD signal change	0.22
α	Steady state flow-volume relation	0.4
β	Field-strength dependent parameter	1.5

Variable	Meaning	
m(t)	$CMRO_2$ normalized to baseline	
f(t)	CBF normalized to baseline	
T(t)	Temperature	
W(t)	Lambert W Function	
$frac\Delta S(t) S_0$	Change in BOLD signal normalized to rest	

Multi-Voxel Approach

2.2 Modeling the BOLD Response

2.3 Modeling Temperature

2.3.1 The Approach

How the temperature is calculated

Calculating the equilibrium temperature

Calculating Metabolism and Blood Flow Changes

Calculating the change in temperature in the active brain

2.3.2 Results

Using Theoretical BOLD Data

Using Experimental BOLD Data

Chapter 3

Detector Applications to measuring the active brain

3.1 Functional Near-Infrared fNIR Imaging

$$I = I_0 e^{-\alpha x} \tag{3.1}$$

3.2 Temperature Measurements

From the Beer-Lambert law eq. (3.1), the penetration depth, δ_p can be expressed as

$$\delta = \frac{1}{\alpha} \tag{3.2}$$

where α is the absorption coefficient. At body temperature (37°) the peak wavelength in the blackbody spectrum is approximately BLA. For water at this wavelength, α is approximately HUGE, so δ is VERY SMALL.
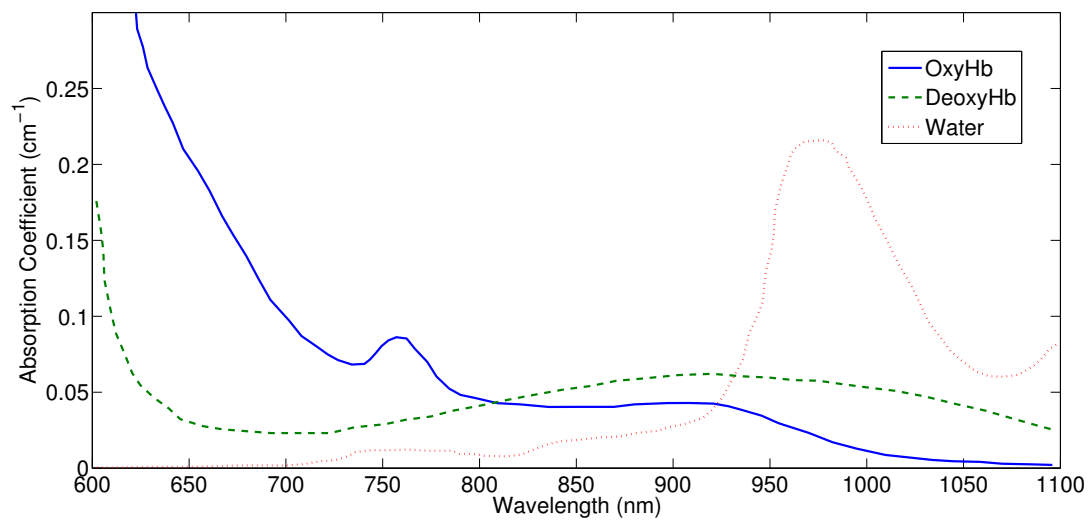
Figure 3.1: Absorption spectra of water, Hb and Dhb. From Cope [6] and HB stuff from Horecker [7]

Chapter 4

Conclusion

Bibliography

[1] Herman Y Carr and E M Purcell. Effects of diffusion on free precession in nuclear magnetic resonance experiments. *Physical Review*, 94:630–638, 1954.

[2] Paul Lauterbur. Image formation by induced local interactions: Examples employing nuclear magnetic resonance. *Nature*, 242:190–191, 1973.

[3] Roberto C Sotero and Nelson J Trujillo-Barreto. Modeling the role of excitatory and inhibitory neuronal activity in the generation of the bold signal. *NeuroImage*, 35:149–165, 2007.

[4] Roberto C Sotero and Yasser Iturria-Medina. From Blood Oxygen Level Dependent (BOLD) signals to brain temperature maps. *Bulletin of Mathematical Biology*, 73(11):2731–2747, 2011.

[5] H H Pennes. Analysis of tissue and arterial blood temperatures in the resting human forearm. *Journal of Applied Physiology*, 1(2):93–122, 1948.

[6] M Cope. *The development of a near infrared spectroscopy system and its application for non invasive monitoring of cerebral blood and tissue oxygenation in the newborn infants*. PhD thesis, 1991.

[7] B L Horecker. The absorption spectra of hemoglobin and its derivatives in the visible and near infra-red regions. *The Journal of Biological Chemistry*, 1942.

Calculating the Temperature Change

```matlab
function temperatureOut = tempCalcChangingMetabolismFlow(tissue,bloodT,
    airT,nt,tmax,pastCalc,start,stop,amplitudeMet,amplitudeFlow,region,
    savesteps)
% tempCalcChaning Metabolism  How does changin metabolism affect things?
%   tissue: holds all of the strucual information
%   bloodT: Temperature of the blood
%   airT:   Temperature of the surrounding ait
%   nt:     Number of time steps
%   tmax:   Total amount of time the simulation should run over
%
%   region: logical matrix same size as head
%   start:  units of steps
%   stop: units of steps
%
%   Writen by Greggory Rothmeier (greggroth@gmail.com)
%   Georgia State University Dept. Physics and Astronomy
%   May, 2011
%#codegen

%%   Default Values
if nargin<2,  bloodT = 37;          end
if nargin<3,  airT = 24;            end
if nargin<4,  nt = 3;               end
if nargin<5,  tmax = 1;             end
if nargin<6,  pastCalc = 0;         end   %  Voxel size (m)
if nargin<7,  start = 10;           end   %  in steps
if nargin<8,  stop = 20;            end   %  in steps
if nargin<9,  amplitudeMet = 1.2;   end   %  normalized
if nargin<10, amplitudeFlow = 1.2;  end   %  normalized
if nargin<11, savesteps = 1;        end

```

```matlab
30
31   dx = 2*10^-3;
32   if nt<(2*tmax),
33       warning('Time step size is not large enough.  Results will be
            unreliable.  Consider increasing the number of steps or reducing
            tmax.')
34   end
35
36
37   % Constants used that aren't already stored in tissue
38   [xmax ymax zmax t] = size(tissue);
39   clear t;
40   dt = tmax/(nt-1);
41   % rhoBlood = 1057;
42   % wBlood = 1000;
43   % cBlood = 3600;
44
45   statusbar = waitbar(0,'Initializing');
46
47   %%  Maps
48   %   Creates a map that identifies where there is tissue
49   %   the condition squeeze(tissue(:,:,:,)~=airIndex picks out the elements
         that are
50   %   tissue
51
52   temperatureOut = ones(ceil((nt-1)/savesteps),xmax,ymax,zmax,'single');
53   temperature = ones(2,xmax,ymax,zmax,'single')*airT;
54   if pastCalc == 0
55       temperature(1,squeeze(tissue(:,:,:,1))~=1) = bloodT;
56   else
57       temperature(1,:,:,:) = pastCalc(end,:,:,:);
58       % temperature(1,:,:,:) = pastCalc;
```

```matlab
59  end
60  temperatureOut(1,:,:,:) = temperature(1,:,:,:);
61  metabMultiplier = ones([xmax ymax zmax],'single');
62  flowMultiplier = ones([xmax ymax zmax],'single');
63
64  %%  Do Work.
65  %   This is a vectorized version of the next section.  For the love of
        god
66  %   don't make any changes to this without first looking below to make
        sure
67  %   you know what you're changing.  This is [nearly] impossible to
68  %   understand, so take your time and don't break it.
69  %   data is stored in 'tissue' as such :
70  %   [tissuetype 0 Qm c rho k w];  <--  second element is blank for all.
71  %   [    1      2 3 4  5  6 7
72
73  %  This makes an array that has smoothed out variations in k by averaging
74  %  all of the k's around each voxel (including itself).  This is a
75  %  hap-hazard solution to the problem that if you only take the value of
        k
76  %  for the voxel without considering what surrounds it, it doesn't matter
77  %  whether the head is surrounded by air, water or anything else.  Since
78  %  water is a better thermal conductor than air, we need a way of
79  %  accounting for this.  This is one way:
80  averagedk = (circshift(tissue(:,:,:,6),[1 0 0])+circshift(tissue(:,:,:,6)
        ,[-1 0 0])+circshift(tissue(:,:,:,6),[0 1 0])+circshift(tissue(:,:,:,6)
        ,[0 -1 0])+circshift(tissue(:,:,:,6),[0 0 1])+circshift(tissue(:,:,:,6)
        ,[0 0 -1])+tissue(:,:,:,6))/7;
81  rhoblood = 1057;
82  cblood = 3600;
83
84  %%  Only saves every 4 steps
```

```matlab
85  tic
86  for t2 = 1:nt-1
87      waitbar(t2/(nt-1),statusbar,sprintf('%d%%',round(t2/(nt-1)*100)));
88      if (start<t2) && (t2<stop)   % for __ steps
89          metabMultiplier(region) = amplitudeMet;   % region is hardcoded
                here
90          flowMultiplier(region) = amplitudeFlow;
91      elseif t2==stop % once the period is over, reset it back to ones
92          metabMultiplier(region) = 1;
93          flowMultiplier(region) = 1;
94      end
95
96      temperature(2,:,:,:) = squeeze(temperature(1,:,:,:)) + ...
97          dt/(tissue(:,:,:,5).*tissue(:,:,:,4)).* ...
98          ((averagedk/dx^2).*...
99          (circshift(squeeze(temperature(1,:,:,:)),[1 0 0])-2*squeeze(
                temperature(1,:,:,:))+circshift(squeeze(temperature(1,:,:,:))
                ,[-1 0 0])+...  % shift along x
100         circshift(squeeze(temperature(1,:,:,:)),[0 1 0])-2*squeeze(
                temperature(1,:,:,:))+circshift(squeeze(temperature(1,:,:,:))
                ,[0 -1 0])+...  % shift along y
101         circshift(squeeze(temperature(1,:,:,:)),[0 0 1])-2*squeeze(
                temperature(1,:,:,:))+circshift(squeeze(temperature(1,:,:,:))
                ,[0 0 -1]))...  % shift along z
102          -(1/6000)*rhoblood*flowMultiplier.*tissue(:,:,:,7)*cblood.*(
                squeeze(temperature(1,:,:,:))-bloodT)+metabMultiplier.*
                tissue(:,:,:,3)));
103     %  resets the air temperature back since it's also modified above,
            but
104     %  it needs to be kept constant throughout the calculations
105     temperature(2,squeeze(tissue(:,:,:,1))==1) = airT;
106     temperatureOut(ceil(t2/savesteps),:,:,:) = temperature(2,:,:,:);
```

13

```matlab
107     temperature(1,:,:,:) = temperature(2,:,:,:);  % moves 2 back to 1
108 end
109 close(statusbar);
110 toc
111 %%  Keeps all of the data
112 %   Note: Make sure that temperature() has enough preallocated space
113 %{
114 tic
115 for t2 = 1:nt-1
116     %tic
117     waitbar(t2/(nt-1),statusbar,sprintf('%d%%',round(t2/(nt-1)*100)));
118     if (start<t2) && (t2<stop)   %  for 10 steps
119         metabMultiplier(region) = amplitudeMet;   %  region is hardcoded
              here
120         flowMultiplier(region) = amplitudeFlow;
121     elseif t2==stop %  once the period is over, reset it back to ones
122         metabMultiplier(region) = 1;
123         flowMultiplier(region) = 1;
124     end
125     temperature(t2+1,:,:,:) = squeeze(temperature(t2,:,:,:)) + ...
126         dt/(tissue(:,:,:,5).*tissue(:,:,:,4)).* ...
127         ((averagedk/dx^2).*...
128         (circshift(squeeze(temperature(t2,:,:,:)),[1 0 0])-2*squeeze(
              temperature(t2,:,:,:))+circshift(squeeze(temperature(t2,:,:,:))
              ,[-1 0 0])+...  % shift along x
129          circshift(squeeze(temperature(t2,:,:,:)),[0 1 0])-2*squeeze(
              temperature(t2,:,:,:))+circshift(squeeze(temperature(t2,:,:,:)
              ),[0 -1 0])+...  % shift along y
130          circshift(squeeze(temperature(t2,:,:,:)),[0 0 1])-2*squeeze(
              temperature(t2,:,:,:))+circshift(squeeze(temperature(t2,:,:,:)
              ),[0 0 -1]))...  % shift along z
```

```matlab
131              -(1/6000)*rhoblood*flowMultiplier.*tissue(:,:,:,7)*cblood.*(
                    squeeze(temperature(t2,:,:,:))-bloodT)+metabMultiplier.*
                    tissue(:,:,:,3));
132      %    resets the air temperature back since it's also modified above,
            but
133      %    it needs to be kept constant throughout the calculations
134      temperature(t2+1,squeeze(tissue(:,:,:,1))==1) = airT;
135      %dispTimeLeft(1,1,nt-1,t2)
136  end
137  close(statusbar);
138  toc
139  %}
140
141  %% Old Code
142  %  This is what used to be used.  It's much slower (~60 times slower),
        but
143  %  it's much easier to understand compared to the above code.  If any
144  %  changes need to be made above, first look through this code to ensure
145  %  you understand what's happening before making changes.  It's really
        easy
146  %  to mess up the code above and nearly impossible to figure out where.
147
148  %  good luck.
149
150
151  %{
152  for t2 = 1:nt-1
153      for x2 = 2:xmax-1
154          for y2 = 2:ymax-1
155              for z2 = 2:zmax-1
156                  if tissue(x2,y2,z2,1) ~= 1,
```

```
157                    temperature(t2+1,x2,y2,z2) = temperature(t2,x2,y2,z2)
                         + (dt/(tissue(x2,y2,z2,5)*tissue(x2,y2,z2,4)))*((
                         tissue(x2,y2,z2,6)/dx^2)*...
158                      (temperature(t2,x2+1,y2,z2)-2*temperature(t2,x2,y2,
                         z2)+temperature(t2,x2-1,y2,z2)+...
159                      temperature(t2,x2,y2+1,z2)-2*temperature(t2,x2,y2,
                         z2)+temperature(t2,x2,y2-1,z2)+...
160                      temperature(t2,x2,y2,z2+1)-2*temperature(t2,x2,y2,
                         z2)+temperature(t2,x2,y2,z2-1))...
161                      -(1/6000)*rhoBlood*wBlood*cBlood*(temperature(t2,x2
                         ,y2,z2)-bloodT)+tissue(x2,y2,z2,3));
162                  end
163                end
164              end
165          end
166      end
167      %}
168
169
170      end
```

Calculating The Equilibrium Temperature