Profile restoration in MASS: fixed layers

A. Tokovinin

Draft 1.0. April 5, 2002

1 Introduction

MASS instrument measures scintillation indices (normal and differential) in 4 concentric apertures. From this data, the vertical distribution of turbulence (profile) is restored. Here we study one particular method of this restoration, when the atmosphere is modeled as a collection of turbulent layers at fixed altitudes. The questions addressed here are:

- What is the real vertical resolution of MASS?
- What is the maximum number of layers that can be restored?
- Can MASS work with only 3 smallest apertures A, B, C?

2 The method

We consider only one turbulent layer of fixed strength and change its altitude from some small initial value to 20 km with a step of 0.125 km. For each layer altitude (or, to be precise, range), a set of true scintillation indices is generated. Only normal (non-generalized) MASS mode is considered, so the set consists of 4 normal and 6 differential indices. Some random noise is added to further simulate real situations. The level of *relative* rms noise corresponds to the actual MASS noise in 1-min exposures: 2% on all indices except AB where the noise is higher, 4%. The noise is practically independent on the brightness of a star, because it is generated by the statistics of scintillation itself.

The set of indices is injected into the restorator, which finds the strength of several fixed layers by minimizing the χ^2. The value of χ^2/N (N being the number of parameters) is a goodness-of-fit measure. Layer strengths are constrained to be non-negative.

The IDL code `massx.pro` is given in the Appendix. Under IDL, you need to do `getweight` first, to read the weighting functions. Then you may type `restoration` to compute profiles for each range of a layer (with independent noise added for each range), and use the command `plotres` to see the results.

3 Results

In Fig. 1 we show the results of the "standard" 5-layer decomposition using all 10 indices with noise. As expected, a layer between, say, 4 and 8 km is represented as a combination of fixed layers at 4 and 8 km, in a relative proportion that depends on the layer position within this interval. This forced redistribution does

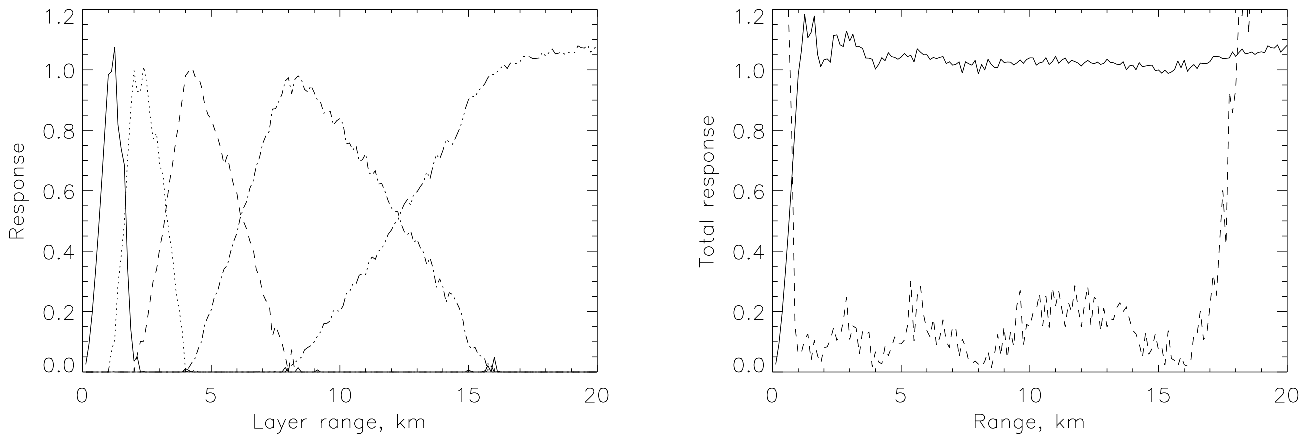

Figure 1: Restoration with full set of indices and 5 fixed layers at 1, 2, 4, 8, 16 km. Left: relative response of each restored layer to a single turbulent layer at a given range. Right: the full line shows the sum of responses, which gives the idea on how well the total intensity is reproduced by restoration. The dotted line is $\chi^2/10N$.

not affect the total energy, as can be seen on the right plot. However, in a situation of maximum mismatch between the model and reality (e.g. when the true layer is between the imposed altitudes) the quality of the fit, χ^2/N, is less good. The influence of noise on reconstruction is seen to be quite moderate, few percent.

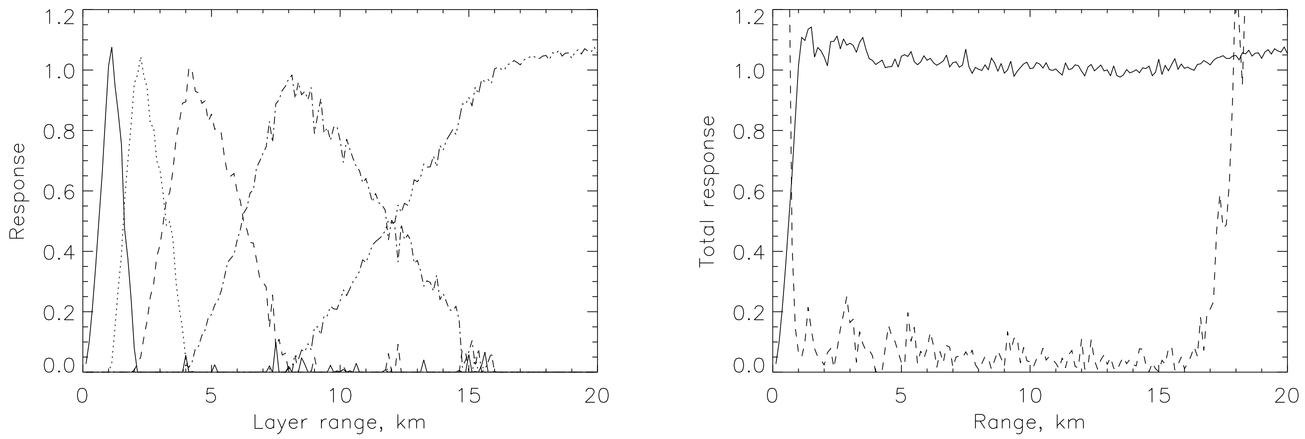

Figure 2: Restoration with reduced set of 6 indices coming from apertures A, B, C and 5 fixed layers at 1, 2, 4, 8, 16 km.

In Fig. 2 the restoration with a reduced set of only 6 indices (aperture D disabled) is shown. This situation may arise when MASS is used on a too small telescope, e.g. off-axis portion of Celestron or Meade. The number of retrieved profile parameters is now only one less than the number of input parameters.

Moreover, the remaining weighting functions are all similar at high altitude, leading to poor discrimination between the high layers. This leads to noise amplification for these layers, as seen in the Figure. Low layers are affected only slightly. Note the "leaking" of power from high layers to lower ones: some 10% of turbulent energy of high layers may supriously appear in the low layers. Similarly, 4 km layers leaks into 1 km layer.

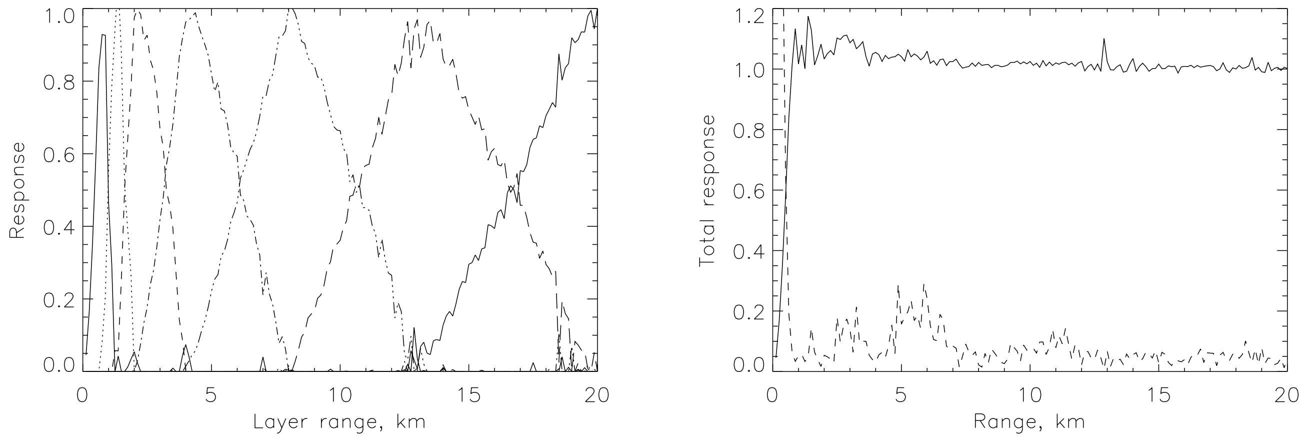

Figure 3: Restoration with full set of indices and 7 fixed layers at 0.7, 1.2, 2, 4, 8, 13, 20 km.

Fig. 2 shows the attempt to increase the number of layers to maximum with a full set of 10 indices. The results are still acceptable, but at some altitudes the energy of a single layer "leaks" again into several non-adjacent restored layers. This cross-talk does not disappear even for zero noise, it is inherent to the problem and reflects the limited vertical resolution of MASS.

A Simulation code massx.pro

```
; Reduction of MASS, fixed-altitude layers
; Apr-05-2002  A.Tokovinin
;------------------------------------------------------------
PRO getweight
; Get the WFs from weight.a05 and save them as wt array
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask
;
tmp = findfile('weight.a05', count=n)
openr, 11, 'weight.a05'  ; open the file for reading
wt = fltarr(11, 27)
tmp = fltarr(11)
s="S"
for i=0,8 do readf, 11, s   ; Skip 9 header lines
FOR i=0,26 DO BEGIN
  readf, 11, tmp
  wt(*,i) = tmp
```

```
ENDFOR
wt(1:10,*) = wt(1:10,*)*1e11
save, filename='weighta5.idl', wt
close, /all
!P.charsize = 1.5
plot, wt(0,*),wt(1,*), xtitle='Height, km'
END
;-----------------------------------------------------
FUNCTION interpweight, h
; returns 10-vector of weights for a given altitude h
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask
j=-1
for i=0,26 do if ((j EQ -1) AND (h LE wt(0,i))) then j=i-1 ; find the closest
if (j EQ -1) then j=25
tmp = wt(1:10,j) + (h-wt(0,j))/ (wt(0,j+1)-wt(0,j))*(wt(1:10,j+1)- wt(1:10,j))
return, tmp
end
;-----------------------------------------------------
PRO prepare, alt, strength
; Prepare the data for layer at alt. ALT with strength STRENGTH
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask

;restore, 'weighta5.idl'  ; heights and weights
;z = [1.4,2.,4.,8.,16.]   ; restoration altitude grid
;z = [1.,2.,4.,8.,16.]
;z = [0.7,1.2,2.,4.,8.,16.]
z = [0.7,1.2,2.,4.,8.,13.,20.]
;mask = [1.,1.,1.,0.,1.,1.,0.,1.,0.,0] ; remove D-aperture
mask = [1.,1.,1.,1.,1.,1.,1.,1.,1.,1.] ; include D-aperture

nl = n_elements(z)
w_mat = fltarr(nl,10)
  for i=0,nl-1 do w_mat(i,*) = interpweight(z(i))*mask

s0 = interpweight(ALT)*STRENGTH ; true indices
ss = s0*mask ; noiseless indices

sigma = 0.02*s0 ; 2% noise
sigma(4)=0.04*s0(4)  ; AB noise increased to 4%
print, 'Indices: ', s0

; for initial approximation: constant profile
xx = strength/nl + z*0.
END
```

```
;-----------------------------------------------------
PRO noise
; Add noise to the true indices
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask
FOR i=0,n_elements(s0)-1 DO ss(i) = s0(i)+randomn(seed)*sigma(i)*mask(i)
END
;-----------------------------------------------------
FUNCTION merit,  y
; Merit finction, y^2=new profile
; input: vector y, y^2 is the profile
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask
s1 = w_mat ## y^2
l = total( (ss-s1)^2/(2.*sigma^2)*mask)
return, l
END
;-----------------------------------------------------
FUNCTION solve
; Restore the profile
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask

y = xx*0. + total(xx)
xi = identity(n_elements(xx))
powell, y,xi,1e-5,fmin,'merit'
chi2 = fmin/total(mask)
print, 'Chi2/N:      ',chi2
return, y^2
END
;-----------------------------------------------------------
PRO result, y
; display the result
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask

s1 = w_mat ## y
print, '  Observed  Modeled Difference'
FOR i=0,n_elements(s0)-1 DO $
  print, format='(3F8.2)', 100.*s0(i), 100.*s1(i), 100.*(s0(i)-s1(i))

int1 = total(xx)*1e13
int2 = total(y^2)*1e13
print, format='("Integrals, e-13 m^(1/3): ",3(F8.3))', int1, int2, (int2-int1)/int1
END
;-----------------------------------------------------
PRO resolution
; Find how the layers are split at restoration
```

```
COMMON mass, z, wt,w_mat, xx, x0, s0, ss, sigma, seed, chi2,mask

strength = 1e-13 ; layer intensity, m^1/3
n = 160          ; number of altitudes
alt = findgen(n)*0.125 +0.125 ; 0.125...20km

prepare, 1., strength ; to define z-grid !
nl = n_elements(z)
res = fltarr(n,nl)
chi2res = fltarr(n)

for i=0,n-1 do begin
  prepare, alt(i), strength
  noise   ; add noise if needed
  res(i,*) = solve()/strength
  chi2res(i)=chi2
endfor

!P.charsize=1.5
plot, alt, res(*,0)
for i=1,nl-1 do  oplot, alt, res(*,i)
save, filename='resolution.idl', alt, res, chi2res
END
;------------------------------------------------------
pro plotres
; plot the restoration results
restore, 'resolution.idl'
nl = n_elements(res(0,*))

; first plot: response functions
plot, alt, res(*,0), xtitle='Layer range, km', ytitle='Response'
for i=1,nl-1 do  oplot, alt, res(*,i), linestyle = i

sum = res(*,0)
for i=1,nl-1 do sum = sum+res(*,i)
stop
; Second plot: sum and Chi2
plot, alt, sum, xtitle='Range, km', ytitle='Total response'
oplot, alt, chi2res/10, linestyle=2
end
;------------------------------------------------------
```