General

As compared to the first version, the paper by Sheng Li, Ke Du has been significantly changed. Many of the concerns I pointed out in the first review have been addressed. I apologize that while reading the new version of the paper, unfortunately, I found some new relevant issues. Some of them are crucial and, to my opinion, should be carefully considered before publication.

In general the language has been greatly improved, however, I would recommend the authors to double check if the editing process has *always* conserved the original meaning of what they wanted to say.

Specific comments

Sect. 2.1: the section has been significantly improved. I see only two remaining issues:

a) Linearity of equation (2). Your statement at line 848 of your replies "*The predicted PIC for one beam equals to the summation of the multiplication of the pixel concentration and the beam length passing it*" holds only in the "optically thin" regime, i.e. only when a small fraction of the laser signal is absorbed by the pollutant to be measured. If the pollutant concentration is large and / or the medium is not very transparent at the laser wavelength, then for sure linearity does not hold. Since the laser wavelength can be tuned, usually it is possible to find a suitable wavelength at which, for average or expected pollutant concentrations, the medium is sufficiently transparent to fall in the linear regime. To my opinion it would be convenient to state explicitly that you are working in this hypothesis.

b) Equation (3): is it really needed to require $\mathbf{c} \geq 0$? Assume that the real concentration of the pollutant to be measured is zero. If you obtain the concentration as the average of several measurements, then, because of the analyzer's noise, these measurements should be evenly distributed around zero to give a zero average value. If you constrain the solution **c** to be greater or equal to zero, then you remove the measurements that otherwise would be negative because of the measurement noise. Thus you introduce a bias (or exposure error) in the average. Of course you don't see this type of error in the tests presented later in the paper, because the test 2D distributions considered do not include a "zero concentration" case, and because all your formulas fully ignore the measurement noise error (that is the noise error on your **b** vector).

Sect. 2.2: If I understand properly, in practice what you call LTD method is the Tikhonov regularization using the discrete 3rd derivative operator $\mathbf{D}_3$. You tune the regularization strength *w* using the discrepancy principle. Then I have a question: you can not introduce the two equations (4) at every pixel of your domain, as you would have problems at the edges of the domain (see index values k+2, k+1 and k-1 in eq. 4). Thus, you can actually introduce only $2N_c$ - 6 additional equations (not $2N_c$ as you state at line 160). If you do so I think you get in trouble if one or more pixels of your domain are not crossed by a ray path. Actually, in this case $\mathbf{L}^t\mathbf{L}$ would be singular and, if also $\mathbf{T}^t\mathbf{T}$ is singular, then you cannot invert the matrix $\mathbf{A}^t\mathbf{WA}$ mentioned at line 172. In conclusion, I think you are using some extra constraints (also involving 0-th order derivatives) at the edges of your domain (as you seem to suggest in the text after eq. 7). Please clarify. Also, what are the symbols *m* and *n* in eq. 7?

Line 243: please explain clearly the equations you use at the edges of the domain, see also the analogous comment above.

Line 245: requiring $\mathbf{c} \geq 0$ introduces a bias (i.e. a systematic exposure error) for very small values of the real concentration distributions **c**, see also the analogous comment above.

Sect. 2.4: Here you define the “true” concentration distributions that will be considered later to assess the various reconstruction methods. You should say also something regarding the synthetic observations **b** that you build on the basis of g(x,y). Which is the spatial mesh (or grid) that you use for the calculation of **b**, starting from g(x,y)? Do you add pseudo-random measurement noise to **b**, to emulate the real world situation in which the instrument is subject to measurement noise? Your equations discard the measurement error. Mathematically this is equivalent to assume uncorrelated noise equal to 1, thus you should add to each element of **b** a pseudo random noise with zero average and standard deviation equal to 1.

Lines 293 and ff: the unusual naming conventions used in this part make it very difficult to understand for me and, I guess, also for the whole atmospheric community. **G** matrices defined at lines 298-300 are usually called “gain matrices”. It is true that in this case they also represent the “generalized inverse” of **L** because the forward model (**Lc**) is linear. Despite of that, I would still call **G** the “gain matrix”, just for uniformity with Rodgers 2000. Equations at lines 299 and 300 should contain a “+” sign within the parenthesis. Matrix **R** is usually called “averaging kernel”, not “resolution matrix”, just because it shows how a change in the real concentration field maps onto the the concentration estimated by the inversion system. The “resolution” is usually a scalar or vector quantifier computed on the basis of **R**. Line 304: what you call “regularization error” is usually called “smoothing error”. Line 305: what is the perturbation error **G** delta_**b** ? Why should the observed quantities be perturbed? Maybe delta_**b** is the measurement error (noise + calibration, etc.)? Please explain. Line 307: note that the non - sensitivity to the “perturbation error” is not always an appreciated property for the solution. In fact this means that the solution is mostly independent from the observed quantity, therefore it mostly depends on the applied constraints, that is exactly what one usually wants to avoid. Eq. 23: why don’t you write explicitly the expression for the Frobenius norm? Which is the advantage of using the Frobenius norm over the more usual trace[**R**] that, according to Rodgers, 2000 would represent the number of degrees of freedom for the signal? If **R** is not a orthogonal matrix (like in this case, I guess) then I am not able to find an easy interpretation for your quantifier of eq.(23).

Line 365 and ff: how do you interpret the systematically slightly worse performance of MC as compared to LTD in terms of exposure error? I am also surprised to see such a large exposure error in the NNLS method as compared to the other two methods. Maybe this due to the larger size of the pixels used with NNLS? In this case it would be better to change eq.(20) by multiplying each concentration by the area of the pixel to which it refers (actually eq.(20) should compare the integrals of the concentration in the considered domain).

Sect. 3.4: at line 144 of the revised paper finally you reveal that you are using the “lsqnonneg” MATLAB routine to minimize the cost functions relating to the various methods considered. I was not able to establish what is exactly the minimization method used by this MATLAB function, however, for sure it does not find the solution mentioned at line 172 of your revised manuscript, because this latter is not bound to be positive. As explained later, I suspect that lsqnonneg is using iterations, which is not optimal at all for the linear case (**b** = **Lc**) you are dealing with. This implies that the run-time analysis presented in sect. 3.4 is applicable only to the unlucky case in which one uses lsqnonneg. The optimal solution of your inversion would be to implement, directly, in a computer program the matrix operations at line 172. This would make the computation time dependent only on Nb (number of measured PICs) and on Nc (number of unknowns). Both Nb and Nc do not depend on the number of sources that you put in your domain, thus the computation time would not depend neither on the number of sources considered nor on their amplitude and location in the domain. You also claim that the MC method is much faster with respect to the LTD because of the fewer constraint equations. To my experience, if the solution at line 172 would be implemented, most of the computation time would be spent in the inversion of matrix $\mathbf{A}^t\mathbf{WA}$, of

dimension Nc. The number of constraint equations used would impact only the calculation of the product $\mathbf{T}^t\mathbf{T}$ (that is needed to compute $\mathbf{A}^t\mathbf{WA}$), thus the total computation time should depend only very marginally on the number of constraints used.
Another discussion point are the absolute values of the computing times shown in Table 4 and their standard deviations: why the computation time changes and thus shows a standard deviation also when the method used and the number of sources do not change? (I suspect this is because lsqnonneg uses iterations whose number changes from test to test). Why is computation time so long? Are you using a very old CPU? Please specify. Assuming your inversion problem dimensions, finding the solution given at line 172 would require less than 1 second with a standard CPU.
In conclusion, finding the solution as $\mathbf{c} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{p}$ (as specified at line 172) would be much faster and the savings in computing time achieved with the MC method over the LTD would be much less important.

Line 378: what do you infer from fig's 4a and 4b ? Are they useful? I was not able to extract information from them. In my previous review I was suggesting to include only Fig.s 4e and 4f. Of course additional figures are welcome, however they should convey useful information that should also be discussed in the text.

Line 388: figures 4e and 4f show diagonal elements significantly smaller than 1, this means that your regularization is actually very strong (*w* is very large). What do you get for trace[**R**] / Nc ? I guess it is << 1. In practice I feel that with a much softer regularization you would get solutions with better accuracy and similar smoothness.

Figure 4: at least, please use the same color scale in the left and right maps, otherwise the comparison of the two methods is very hard ....

Section 3.6: the results reported in this section are not useful as they are not general for the experimental setup considered, they depend on the dimension of the assumed sources. From what stated at line 399, I understand that these sensitivity tests are performed assuming 5 Gaussian source functions, whose minimal Full Width at Half Maximum (FWHM) is 2.8 m * 2.355 = 6.59 m (see line 272 and the properties of the Gaussian function). When the dimension of the smallest source function is about 6.6 m, clearly there is no much advantage to go from a 18x18 grid (pixel size = 40m / 18 = 2.22 m) to a 40 x 40 grid (pixel size = 40m / 30 = 1.33 m) as in both cases the pixel size is significantly smaller than the source size. Usually, in real situations, the size of the source is unknown and what is useful to know is the spatial resolution of the measuring plus inversion system. The averaging kernels are usually an adequate tool to answer this question: the dimension of the main yellow spots of maps in fig. 4c and 4d quantify the spatial resolution of your measuring system at the location of indexes 16 and 4.

Lines 410 – 413: from the theoretical point of view, the growth of computation time for this inverse problem should roughly follow a polynomial law as a function of the number of pixels. Did you really prove that the behavior you find is exponential? If the behavior is really exponential then you should give a justification so as why the computation time increases so rapidly in your case.

Conclusion section: as outlined above, in my view the time computing savings of the MC method, if any, should be assessed using an optimal solving algorithm for your inversion problem, not the MATLAB lsqnonneg routine. In conclusion, either you renounce claiming the time savings or you implement an optimized solver (like eq. at line 172 implemented in FORTRAN or in C).

Minor corrections

Line 14: I would say “regularization term” (regularization is not a factor in the inversion formula)
Line 22 (and line 113): It is also simpler to perform… maybe you meant “implement” instead of “perform” ?
Line 204: bending ?
Line 272: units of sigma_x and sigma_y should be “m” (meters).
Line 416: seminors ?