

## MeqParm: Parameter Handling in the MeqTree System

M. Mevius, O.M. Smirnov, J.E. Noordam

*ASTRON, P.O. Box 2, 7990AA Dwingeloo, The Netherlands*

**Abstract.** The MeqTree module provides a flexible system to implement arbitrary Measurement Equations and to solve for (subsets of) its parameters. Within the framework of MeqTree, the MeqParms are the (solvable) parameters of the Measurement Equation. The MeqParms represent functions, the coefficients of which are the parameters that are in fact solved for. Examples of MeqParms are discussed, as well as methods for parameter bookkeeping and evaluation, and tools for online inspection and adjustment.

### 1. Introduction

The MeqTree module provides a flexible tool for solving subsets of parameters of arbitrary Measurement Equations (M.E.) (Smirnov & Noordam, this volume). The module is designed to cope with the challenges of calibration of the next generation radio interferometers. Its design is tree/node based, each node class representing a small step in the total calculation of the tree, mostly simple mathematical operations. A node passes requests to its children, receives their values on the requested domain, performs its operation and passes the result to its parent.

A special class of *leaf* nodes (a node without children) is the MeqParm. The MeqParms are the (solvable) parameters of the M.E. A MeqParm represents a function that can be dependent on any axis of variability. The parameters can correspond to instrumental and other effects but can also define a source model. The Local and Global Sky Model (a problem specific, respectively all sky database) (Smirnov& Noordam, 2004 and Nijboer, Noordam & Yatawatta, this volume ) can be implemented using MeqParms. This offers the possibility to include more sophisticated sources and time/frequency variability.

This paper gives a description of the current implementation and the possibilities of the MeqParm.

### 2. Funklets

A parameter node in the MeqTree system represents a function  $f(x_0, \dots, x_n | c_0, \dots, c_m)$  on a given domain for  $x_i$ . The coefficients  $c_i$  are the parameters that, in case of a solvable MeqParm, are in fact solved for. The functions represented by a MeqParm will be denoted *funklets*. The default funklet is a two dimensional

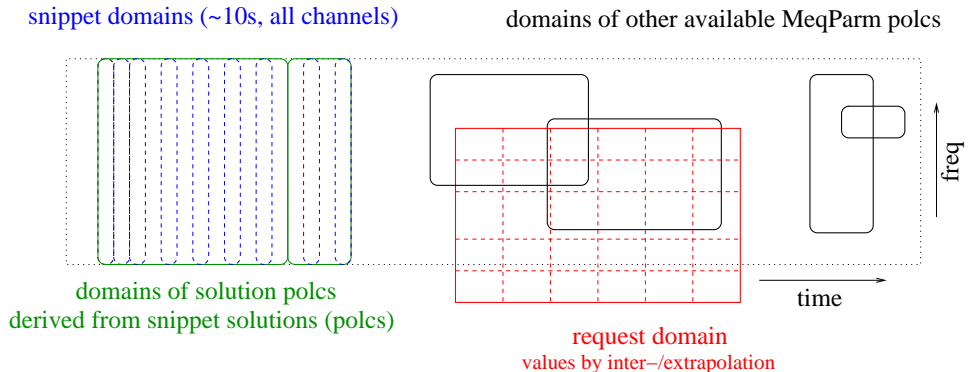


Figure 1. Schematic representation of a ParmTable, with several funklets available for different domains.

polynomial of arbitrary order in frequency and time:

$$f(f, t) = c_{00} + c_{10} \cdot f + c_{01} \cdot t + c_{11} \cdot f \cdot t + c_{20} \cdot f^2 + \dots \quad (1)$$

Several predefined funklets are available, but in principle any other (real) function can be defined with the help of the Aips++ Functional class, which has the possibility to interpret strings as functions. Functions of large complexity can be described by a single funklet. A more sophisticated example is a 'beamshape' model: a two dimensional Gaussian with its coefficients polynomials in frequency and time:

$$g(l, m) = \exp(-(f_{00} + f_{10} \cdot l + f_{01} \cdot m + f_{11} \cdot l \cdot m + f_{20} \cdot l^2 + f_{02} \cdot m^2)), \quad (2)$$

with each  $f_{ij}$  a frequency/time polynomial as defined above. The Python based interface to the MeqTree system eases the definition of such funklets considerably.

### 3. Parameter Evaluation

All information of the funklet can be stored in and retrieved from (Aips++) tables (ParmTable). A funklet is valid for a specific domain and several funklets can belong to a single MeqParm, eg. because independent solutions were generated for different subdomains (Fig. 1). Different domains can have different types of funklets. Upon receiving a request for a specific domain the MeqParm always returns a values. If no suitable funklet is found in the connected ParmTable, the best solution should be determined from either a default funklet or via inter-/extrapolation. Inter- and extrapolation routines can be especially important in calibration runs, where calibration data is taken before and/or after the actual measurement.

If there are several funklets available, each one valid on different subdomains of the larger request domain, the MeqParm is evaluated separately on the subdomains. The user is also able to specify this by dividing its solvedomain into regular tiles. In this case, a solvable MeqParm will have an independent solution

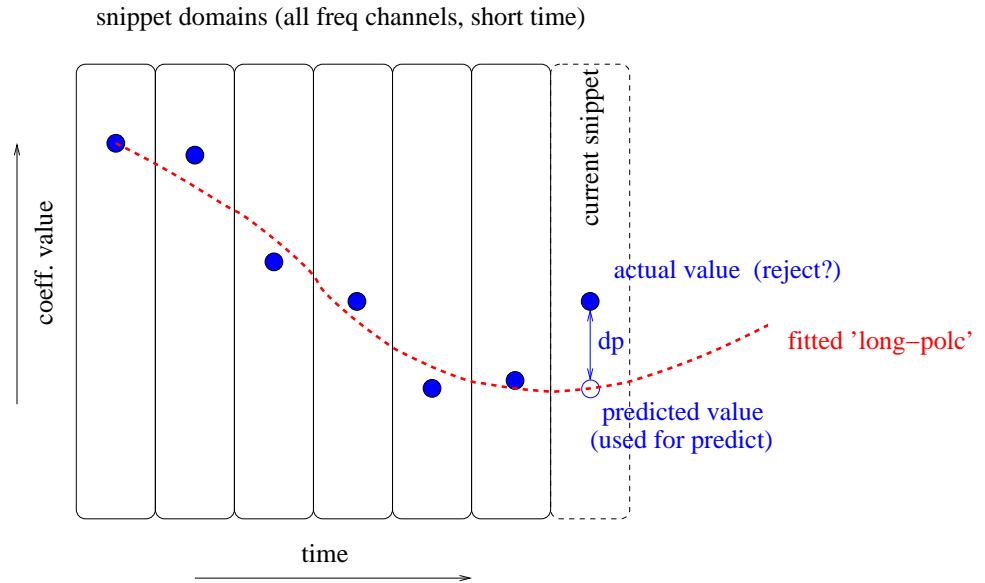


Figure 2. An example in which a MeqParm receives requests for subsequent time slots. The funklet’s coefficients for the next time slot are predicted by a linear fit to the previous values. The information of the difference between predicted and actual value can be used to discard outliers.

for every tile. This offers the user maximum flexibility in deciding the strategy to solve complex problems on large domains. Since all information is stored, a fit on the several independently solved subdomains can always be performed afterwards.

In the case of solvable parameters not only the values of the MeqParm are determined for a requested domain, but also perturbed values for all coefficients of the funklet(s). These perturbed values are transported down through the tree, to the point where they are used to determine the derivatives of the condition equations.

In non-linear problems, many iterations are needed to converge to the right solution. The number of iterations can be considerably reduced by smart initialization of the MeqParm. An example is shown schematically in Fig. 2, where requests are generated for subsequent time slots. By fitting the variation of the coefficient values over time, the next value can predicted. This method can also be used to trace and discard outliers.

#### 4. Online Tools

The Python based GUI (*meqbrowser*) that is part of the MeqTree module, allows for quick inspection/visualization of any node at any stage of the solve-process. Within the browser two special tools have been developed for the MeqParm. One tool provides a graphical display of the contents of the ParmTable, i.e. the solutions of a certain parameter at different validity domains. The other,

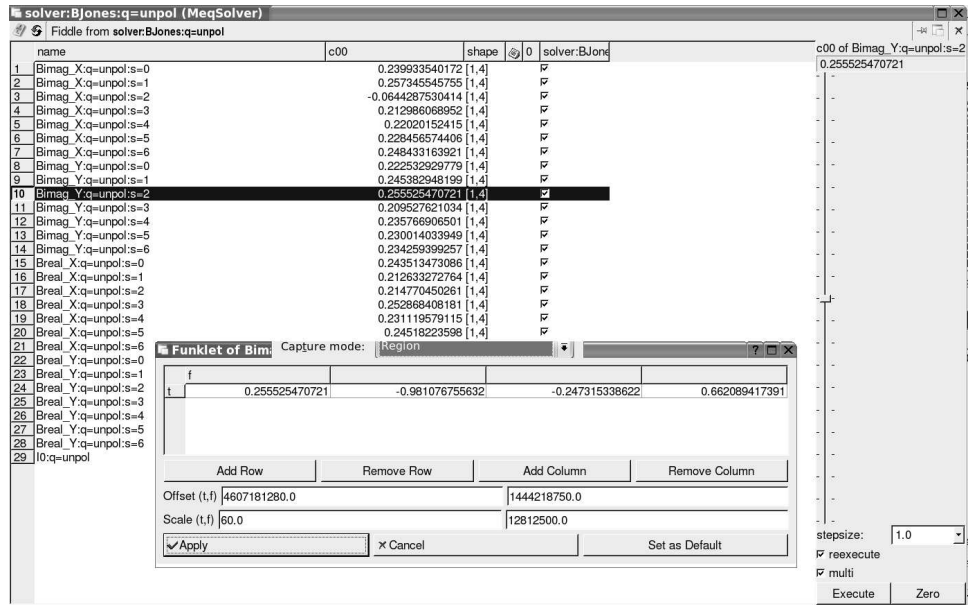


Figure 3. Parameter Fiddling Tool. Lists all parameters under a certain node, allows for adjusting them and also offers the possibility to reexecute the node, to inspect the effects of the changes made.

the parameter adjustment tool, with which it is possible to change online the solvability, coefficient values etc., is especially useful in the design and tests of complex trees.

## References

- Smirnov, O.M., Noordam, J.E. 2004, in ASP Conf. Ser., Vol. 314, ADASS XII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP)
- Smirnov, O.M., Noordam, J.E. 2006, this volume, [P.173]
- Nijboer, R.J., Noordam, J.E. & Yatawatta, S. 2006, this volume, [P.57]