

Radioastronomische Beobachtungstechniken

Schülerpraktikum von David Röser (14.2-25.2.2011)

Radioteleskope befinden sich an vielen Standorten auf der Welt, so auch in Bad Münstereifel-Effelsberg.

Dieses Teleskop ist mit einem Schüsseldurchmesser von 100m das 2. größte frei bewegliche der Welt (Das größte frei bewegliche Radioteleskop ist das *Robert C. Byrd Green Bank Telescope* in den USA). Das größte Radioteleskop überhaupt, welches aber in den Boden eingelassen ist und somit nur im Zenit messen kann, ist das Arecibo-Teleskop (in Puerto Rico) mit einem Schüsseldurchmesser von etwa 300m.

Der Zweck eines Radioteleskopes besteht darin, die von interstellaren Medien, oder Molekülwolken ausgesandte Strahlung im Radiobereich (Wellenlängen im Millimeter- bis Meterbereich) an bestimmten Stellen oder in bestimmten Regionen aus dem Universum zu empfangen, um daraus auf chemische Eigenschaften, Magnetfelder, Temperatur und Geschwindigkeit zu schließen.

Zur Verbesserung der räumlichen Auflösung und Positionsgenauigkeit können viele Radioteleskope weltweit zu einem großen „zusammengeschaltet“ werden. Dieses wird VLBI (=Very Long Baseline Interferometry) genannt, weil es auf dem Prinzip der Interferometrie aufbaut und im Prinzip wie ein Doppelspaltexperiment „rückwärts“ funktioniert:

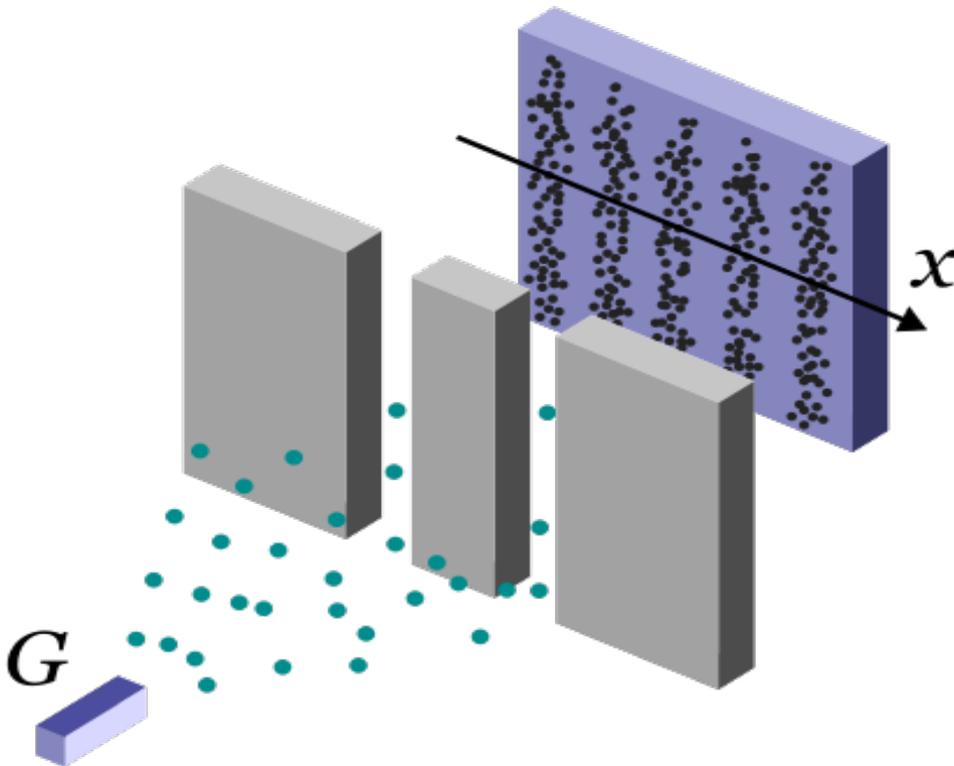


Abb. 1: Doppelspaltexperiment (copyright wikimedia)

Beim Doppelspaltexperiment kann von der Position der Strahlungsquelle, zum Beispiel einer Lichtquelle, auf die Position der Lichtflecken geschlossen werden. Hier wird nun von den „Strahlungsflecken“ die exakte Position der Quelle errechnet.

Die große räumliche Positionsgenauigkeit ist zum Beispiel für die Festlegung astronomischer Koordinatensysteme, aber auch für irdische geodätische Messungen (z.B. Kontinentaldrift) wichtig.

Ein „Zusammenschalten“ vieler Radioteleskope in ganz Europa ist nicht ohne weiteres möglich, da die Photonen auf ihrem Weg zu den Teleskopen unterschiedliche Entfernungen zurücklegen (dieser Abstand ist in der Grafik mit „Physical Delay“ beschrieben, für die die Strahlung, die eine Geschwindigkeit hat, die der Lichtgeschwindigkeit entspricht ($c=299792458$ meter/sekunde), unterschiedliche Zeiten braucht. Diese Zeitverzögerung muss korreliert werden:

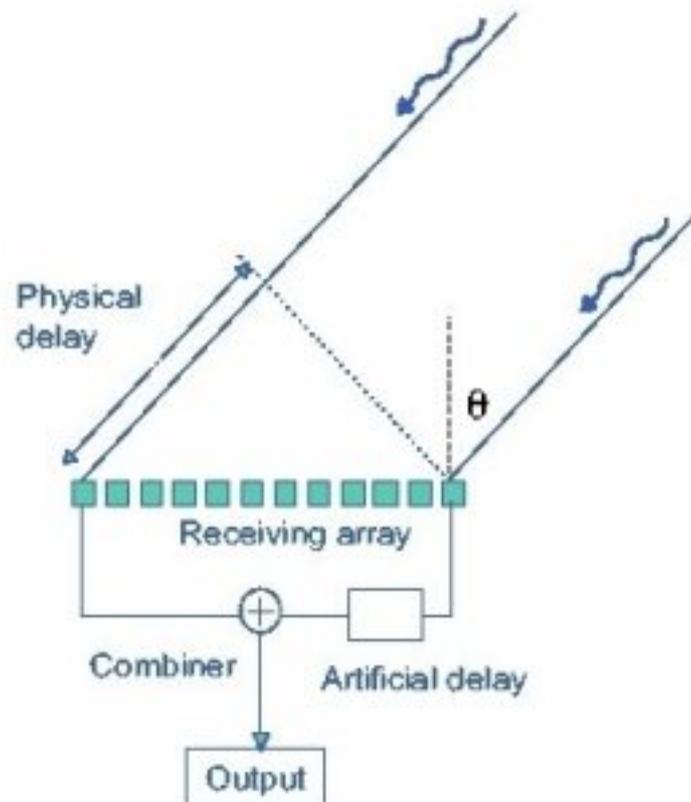


Abb. 2: Strahlengang beim VLBI

Wie wird gemessen?

Um eine exakte Ausrichtung des Teleskopes garantieren zu können, wird zunächst ein Pointing gemacht. Hierbei wird das Teleskop in Elevations-Richtung (vertikale Richtung) über die Quelle gefahren, bis die Amplitude (Größe des Ausschlags der Messgeräte) ihren Höhepunkt erreicht. Danach wird auf die gleiche Weise in Azimut-Richtung (horizontale Richtung) verfahren.

```

52; 1 W3OH      NONE      P10MM01-PB01 O:16-FEB-2011 R:16-FEB-2011
RA: 02:23:16.45 DEC: 61:38:58.0 Eq 1950.0 Offs: -6.1E+4 -2.2E+5
Unknown tau: 0.000 Tsys: -2. Time: 0.37 min El: 59.7
N: 352 ID: 176.0 AO: -0.5089 Da: -8.3989E-03
FD: 30876.0000 Df: 10.00 Pos. Ang. : 0.00 Eq

```

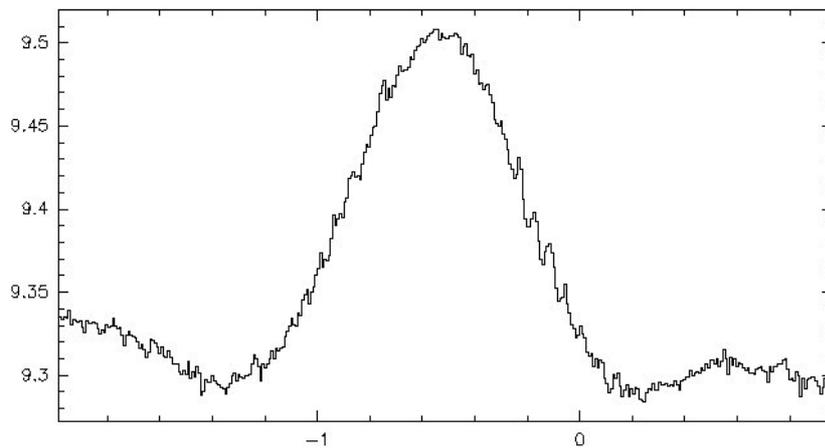


Abb. 3: Messwerte bei einem Pointing:

Um genaue Beobachtungen anzustellen, müssen, da Strahlung hier auf der Erde nur in geringer Intensität ankommen, ausreichend viele Photonen (=Strahlungsquanten) eingefangen werden. Dafür ist es notwendig, eine große Schüssel zu konstruieren, die die hier ankommenden Wellen auf Empfänger reflektiert, die sich entweder in der Primär- oder Sekundärkabine befinden. Diese Empfänger liefern als Messergebnisse Spektren, die angeben, welche Wellen mit welcher Frequenz wie oft detektiert worden sind.

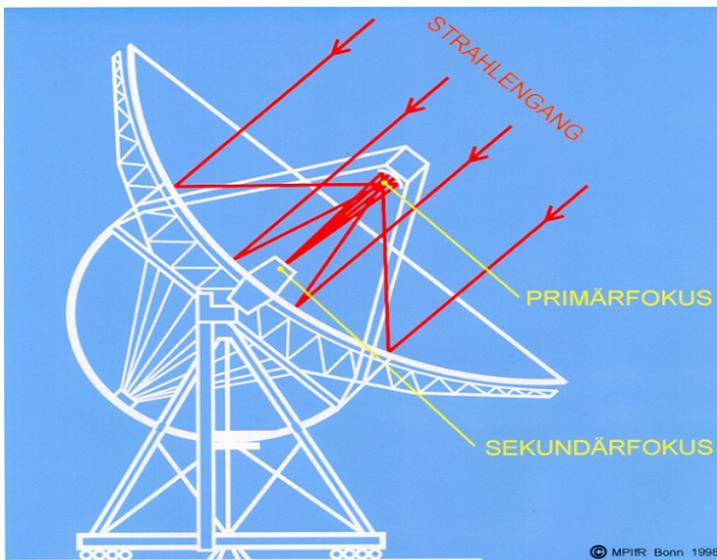


Abb. 4: Aufbau des Radioteleskopes Effelsberg

Hier lässt sich eine Parallele zu Prismen ziehen, die Licht (=Wellen im optischen Bereich) je nach Farbe (=Wellenlänge) verschieden brechen, sodass ebenfalls ein Spektrum entsteht. An diesem Spektrum lässt sich ebenfalls ablesen, welche Frequenzen im Licht vorkommen (=welche Farben vorkommen) und wie intensiv einzelne Strahlen vertreten sind (=wie hell die einzelnen Farben leuchten).

Luftfeuchtigkeit spielt bei Messungen mit niedriger Wellenlänge (= hohe Frequenz) eine große Rolle, da der „Mikrowellen-Effekt“ eintritt (Mikrowellen-Effekt heißt, dass Wellen mit hoher Frequenz von Wassermolekülen absorbiert werden, die sich dadurch in einem angeregten Zustand befinden). Diese Absorption verhindert, dass die Photonen von einem Radioteleskop detektiert werden. Bei zu großer Wellenlänge wird die Strahlung von der Ionosphäre (=Schicht vieler elektrisch geladener Teilchen in der Atmosphäre) zurück ins Universum reflektiert. Die Länge der Radiowellen liegt so, dass sie durch die Atmosphäre dringen können:

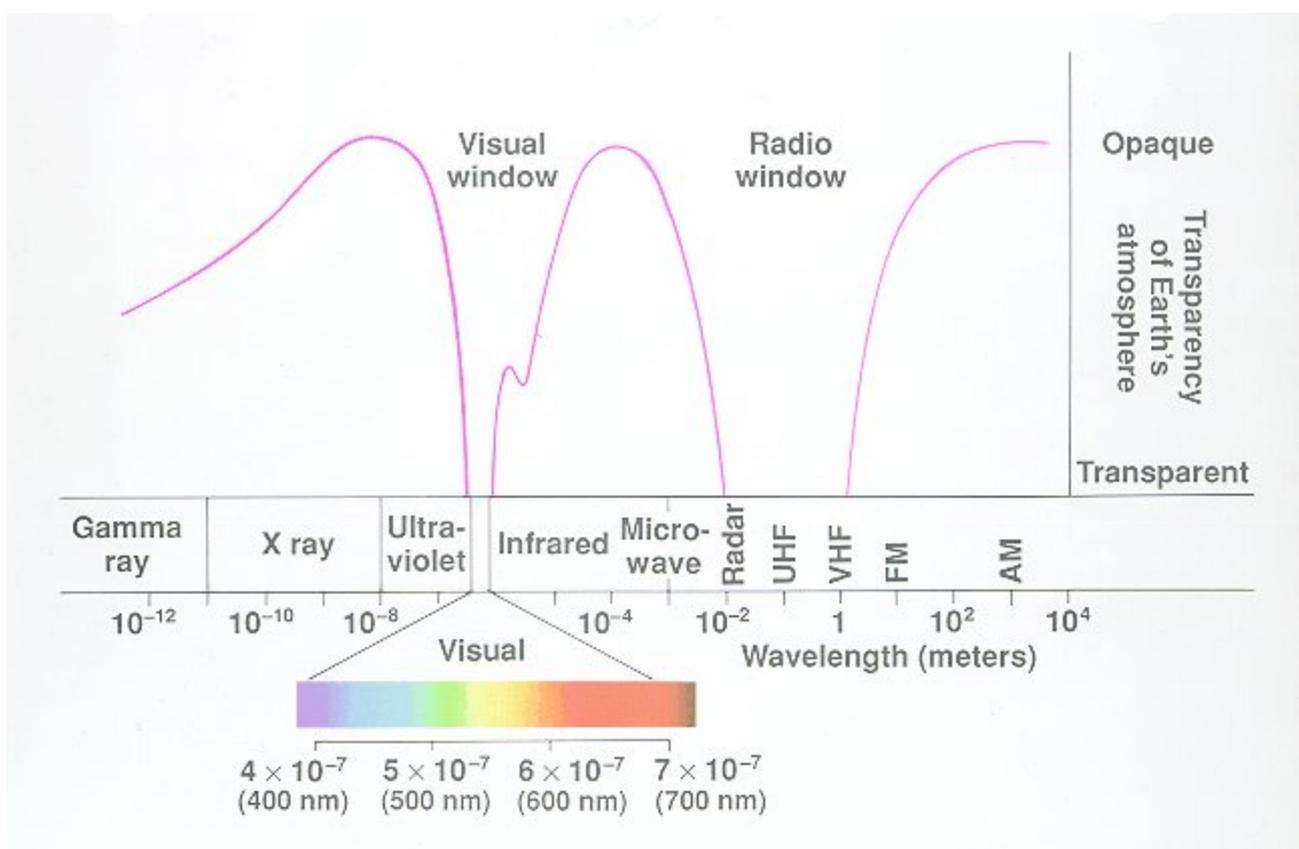


Abb. 5: Welche Wellen dringen durch die Atmosphäre?

Allerdings stammt aber nicht jede von einem Radioteleskop empfangene Strahlung aus dem Universum. Zum Beispiel Handys oder WLAN-Router emittieren Strahlung, die von Teleskop wahrgenommen wird. Solche Quellen lassen sich noch in einem Abstand wahrnehmen, der der Entfernung Erde-Mond entspricht.

Da solche Wellen nicht durch große Objekte dringen können, wurde der Standort des Teleskops in ein von Bergen geschütztes Tal verlegt.

Allerdings ließ sich das Problem der Störquellen durch diese Maßnahme nicht komplett lösen,

sodass trotzdem Referenzmessungen durchgeführt werden müssen.

Als Referenzmessung wird nun eine Stelle am Himmel in der Nähe der On-Position gewählt, an der dennoch keine im Universum befindliche Strahlungsquelle erwartet wird (die sogenannte Off-Position), sodass hier nur störende Wellen empfangen werden:

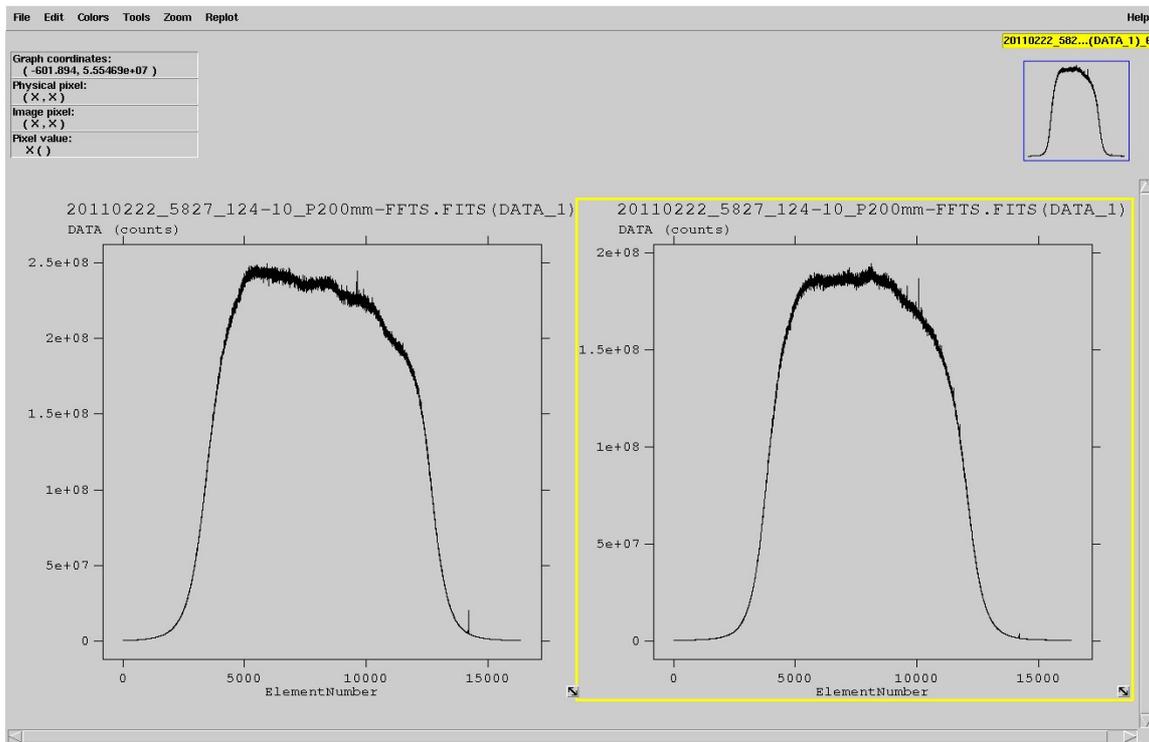


Abb. 6: Off-Position

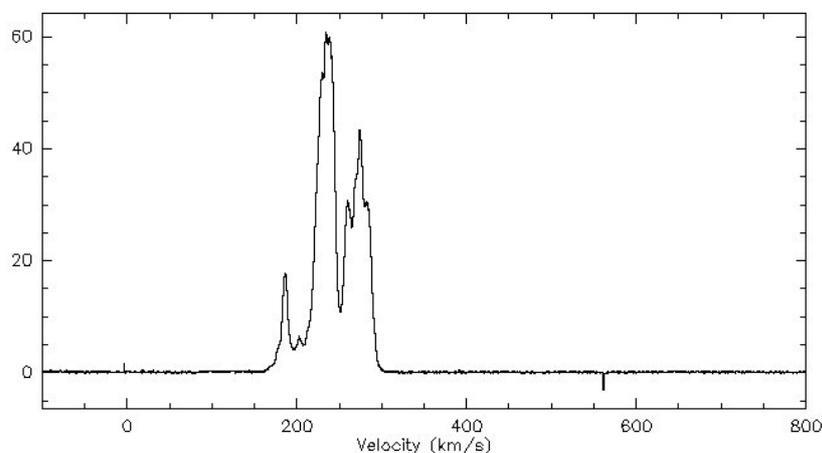
On-Position

Verrechnet man nun die Messergebnisse der Off-Position von denen der On-Position bleibt fast nur das Signal übrig, das tatsächlich von außerhalb der Erdatmosphäre stammen. Ausgenommen ist hierbei Strahlung, die zum Beispiel vom Teleskop selbst emittiert werden.

```

716; 1 S7      NONE      P200M01-FF02 Q:17-FEB-2011 R:17-FEB-2011
l: 132.000 b: -1.000 Ga Offs: +1.5 +0.4
Unknown tau: 0.000 Tsys: 19. Time: 1.5 min El: 26.9
N: 16384 l0: 8192.50 V0: -50.00 Dv: -0.2576 LSR
F0: 1420.40575 Df: 1.2207E-03 Fi: 1420.40675
    
```

Abb. 7:
eigentliches
Messergebnis
Spektrale
Emission des
atomaren



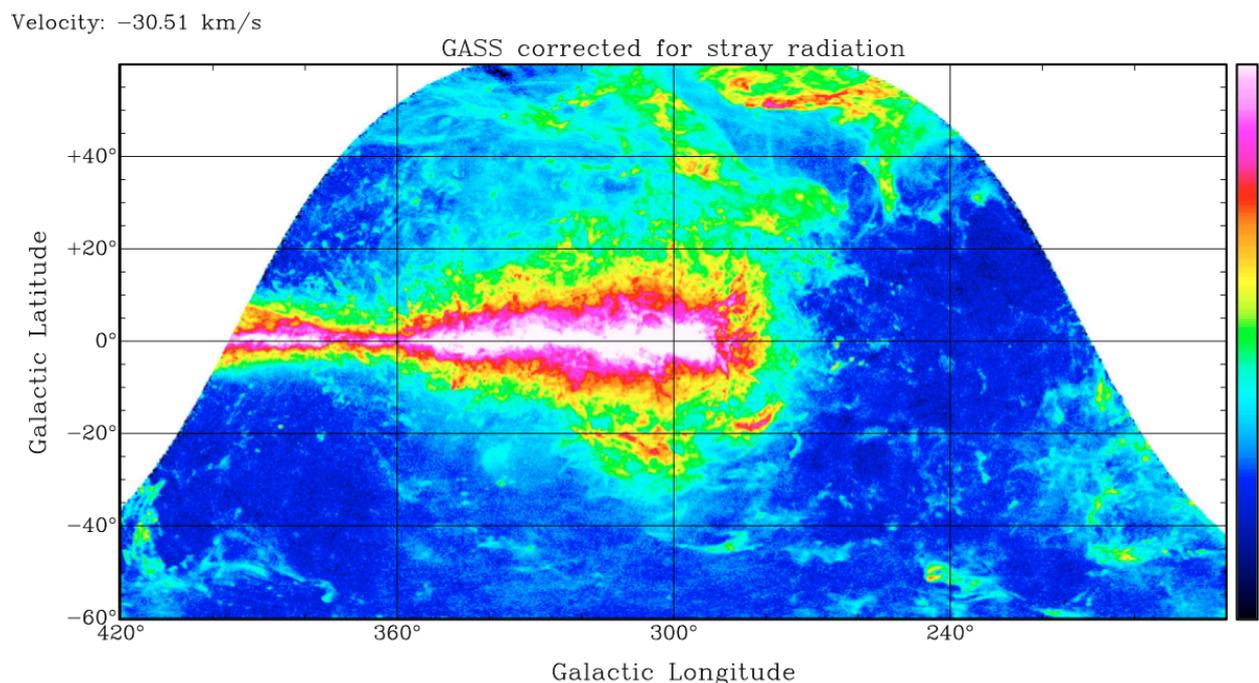
Wasserstoffgases in der Milchstraße bei einer Wellenlänge von 21cm gemessen mit dem 100m Radioteleskop Effelsberg.

Karten Erstellen

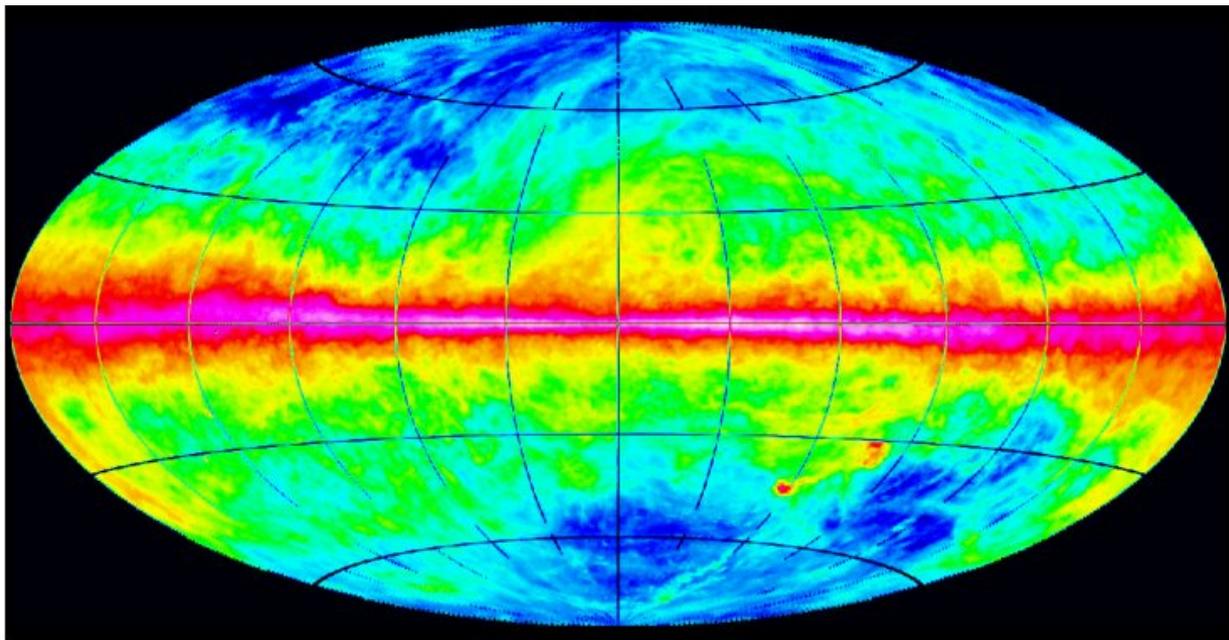
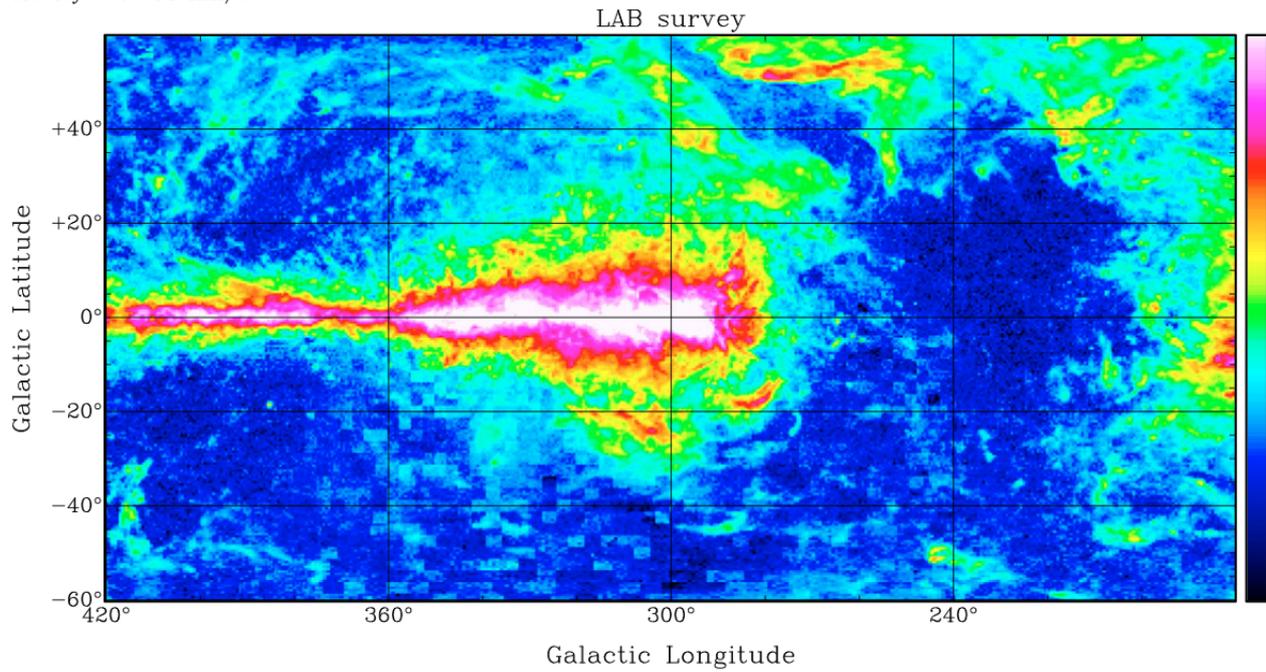
Um zu einem detailreichen Bild eines großen Himmelsobjektes zu kommen, werden von einem Objekt an verschiedenen Punkten Daten gesammelt, die dann entsprechend ihren galaktischen oder intergalaktischen Koordinaten in ein Koordinatensystem eingetragen werden. Eine solche Karte gibt die Intensität der Strahlung bei einer gewissen Frequenz, bzw. einer relativen Geschwindigkeit (die Erde bewegt sich um die Sonne), mit der sich dieses Objekt von der Erde weg oder auf sie zu bewegt, an. Solche Karten werden nicht nur von großen Gebieten (z.B. der Milchstraße) erstellt, sondern auch von einzelnen Gaswolken (z.B. W3MAIN, ein Sternentstehungsgebiet in der Milchstraße).

Auf der x- und y-Achse sind die galaktischen Koordinaten aufgetragen, die Intensität der Strahlung wird durch unterschiedliche Farben dargestellt.

Hier nun ein Paar Beispiele von Karten unserer Milchstraße:



Velocity: -30.44 km/s



(copyright lab-survey)

Die letzten beiden Karten, während der LAB-Survey (=Leiden/Argentine/Bonn) entstanden, zeigen Ansichten der Milchstraße bei der Wellenlänge von 21 cm, die charakteristisch für H I-Wasserstoff ist. Letztere ist eine Komplettansicht mit dem galaktischen Zentrum in der Mitte. Die erste Karte zeigt den Himmel, der von der Südhalbkugel einsehbar ist. Sie ist bei der GASS-Survey (Galactic All Sky Survey) entstanden, der neuen exakteren Kartografierung der gesamten Milchstraße.

Gauß-Fitting

Um zum Beispiel bei einem Pointing die Amplitudengröße und Halbwertsbreite exakt bestimmen zu können, ist es notwendig, die mit dem Teleskop aufgenommenen Messwerte durch eine Funktionsgleichung zu beschreiben. Dies geschieht für ein Pointing zum Beispiel am besten durch eine Gauß-Funktion (dieser Vorgang wird Fitting genannt):

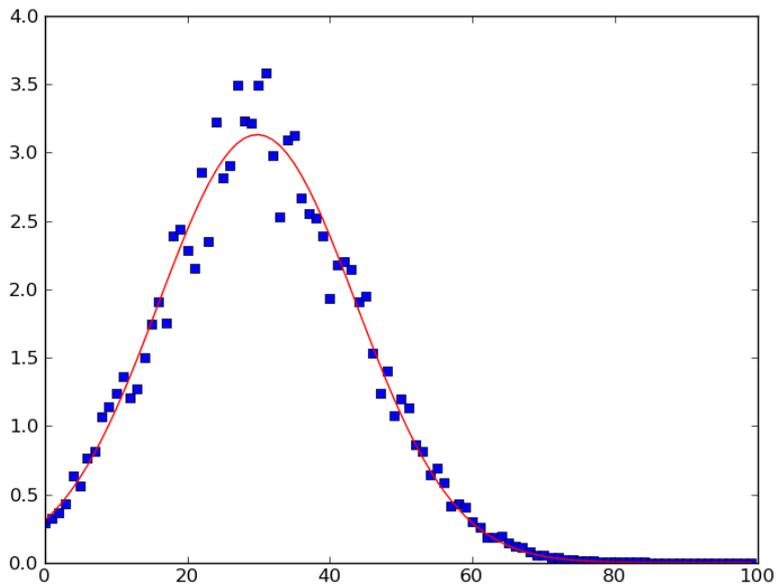


Abb. 8: angefitte Gaußfunktion

Hier ist das Messergebnis durch blaue Punkte dargestellt, durch die die angefitte Gauß-Kurve verläuft. Es war eine meiner Aufgaben während des Praktikums, ein Programm (in der Programmiersprache Python) zu schreiben, das Messwerte durch eine möglichst passende Funktionsgleichung beschreibt:

```
##### curve_fit
import pylab
import os
import math
import sys
import numpy
import scipy
import matplotlib.pyplot as plt
from scipy.optimize import leastsq

def _general_function(params, xdata, ydata, function):
    return function(xdata, *params) - ydata

def _weighted_general_function(params, xdata, ydata, function, weights):
    return weights * (function(xdata, *params) - ydata)
```

```
def curve_fit(f, xdata, ydata, p0=None, sigma=None, **kw):
```

```
    """ Use non-linear least squares to fit a function, f, to data.
        Assumes ``ydata = f(xdata, *params) + eps``
```

Parameters

```
f :          callable
            The model function, f(x, ...). It must take the independent
            variable as the first argument and the parameters to fit as
            separate remaining arguments.

xdata :      An N-length sequence or an (k,N)-shaped array
            for functions with k predictors.
            The independent variable where the data is measured.

ydata :      N-length sequence
            The dependent data --- nominally f(xdata, ...)

p0 :         None, scalar, or M-length sequence
            Initial guess for the parameters. If None, then the initial
            values will all be 1 (if the number of parameters for the function
            can be determined using introspection, otherwise a ValueError
            is raised).

sigma :      None or N-length sequence
            If not None, it represents the standard-deviation of ydata.
            This vector, if given, will be used as weights in the
            least-squares problem.
```

Returns

```
popt :      array
            Optimal values for the parameters so that the sum of the squared error
            of ``f(xdata, *popt) - ydata`` is minimized

pcov :      2d array
            The estimated covariance of popopt. The diagonals provide the variance
            of the parameter estimate.
```

Notes

The algorithm uses the Levenburg-Marquardt algorithm:
scipy.optimize.leastsq. Additional keyword arguments are passed directly
to that algorithm.

```
"""
```

```
if p0 is None or numpy.isscalar(p0):
    # determine number of parameters by inspecting the function
    import inspect
    args, varargs, varkw, defaults = inspect.getargspec(f)
    if len(args) < 2:
```

```

        msg = "Unable to determine number of fit parameters."
        raise ValueError(msg)
    if p0 is None:
        p0 = 1.0
    p0 = [p0]*(len(args)-1)

    args = (xdata, ydata, f)
    if sigma is None:
        func = _general_function
    else:
        func = _weighted_general_function
        args += (1.0/asarray(sigma),)
    res = leastsq(func, p0, args=args, full_output=1, **kw)
    (popt, pcov, infodict, errmsg, ier) = res

    if ier not in [1,2,3,4]:
        msg = "Optimal parameters not found: " + errmsg
        raise RuntimeError(msg)

    if (len(ydata) > len(p0)) and pcov is not None:
        s_sq = (func(popt, *args)**2).sum()/(len(ydata)-len(p0))
        pcov = pcov * s_sq
    else:
        pcov = [[999999.9,0.0,0.0,0.0], [0.0,999999.9,0.0,0.0], [0.0,0.0,999999.9,0.0],
                [0.0,0.0,0.0,999999.9]]
    return pop, pcov

```

Hier folgt nun die Funktionsgleichung einer Gauß-Funktion (ebenfalls in der Programmiersprache Python geschrieben):

```

def gaussian(height, center_x, width_x):
    width_x = float(width_x)
    return lambda x: height*exp(-(((center_x-x)/width_x)**2))

```

Um die Tauglichkeit dieses Programmes zu beweisen, wird künstlich ein Datensatz erzeugt. Dabei werden für eine Reihe von x-Werten die zugehörigen y-Werte ausgerechnet, die dann noch, um Rauschen zu simulieren, mit einer Zufallszahl zwischen 0.8 und 1.2 multipliziert werden. So wird das Ergebnis in einer Weise verfälscht, wie es bei einer echten Messung (fast) immer der Fall ist:

```

xdata=np.array(range(100),'d')
data = gaussian(3, 30,20)(xdata)
for i in range(len(data)):
    data[i]=data[i]*random.uniform(0.8,1.2)

```

