

Eidgenössische Technische Hochschule Zürich

Institut für Geodäsie  
und Photogrammetrie

Bericht

**234**

**Mikrowellen-Wasserdampf-Radiometrie  
und deren Anwendung  
in der Geodäsie**

**Teil I**

**Verfahren,  
Hard- und Softwarebeschreibung**

**Beat Bürki  
Hans Hirter  
Marc Cocard  
Hans-Gert Kahle**

März 1994

**Anmerkung des Herausgebers**

Die Beiträge, die in der Schriftenreihe "Berichte des Instituts für Geodäsie und Photogrammetrie" erscheinen, dienen vor allem dem Unterricht und der Dokumentation. Sie sind deshalb in erster Linie für Mitarbeiter des Instituts und für Studenten bestimmt. Einzelne Hefte können auch einem weiteren Kreis von Interessenten zur Verfügung gestellt werden. Die Auflage ist auf den besonderen Zweck des Heftes abgestimmt.

**IGP Bericht Nr. 234**

**Mikrowellen-Wasserdampf-Radiometrie und deren  
Anwendung in der Geodäsie  
Teil 1: Verfahren, Hard- und Softwarebeschreibung**

**Beat Bürki  
Hans Hirter  
Marc Cocard  
Hans-Gert Kahle**

**© 1994  
Institut für Geodäsie und Photogrammetrie  
an der Eidg. Technischen Hochschule Zürich**

**Alle Rechte vorbehalten**

**ISBN 3-906513-53-X**

## VORWORT

Der vorliegende Band Nr. 234 des Instituts für Geodäsie und Photogrammetrie (IGP) ist ein erster Zwischenbericht im Rahmen des ETH Forschungsprojektes

### **Entwicklung von Mikrowellen-Wasserdampfradiometern für geodätische Satellitenmessungen mit GPS**

Die Auswertung der im UHF-Bereich ausgestrahlten GPS-Satellitensignale erzielt so hohe Genauigkeiten, dass die Abklärung und Modellierung von systematischen und stochastischen Fehleranteilen, die durch die Troposphäre und Ionosphäre verursacht werden, von massgebender Bedeutung geworden sind. Besondere Aufmerksamkeit soll dabei auch den Bedürfnissen der Praxis und Landesvermessung geschenkt werden, die möglichst feldtaugliche und operationelle Lösungsansätze brauchen. In den Arbeiten von ECKERT, COCARD und GEIGER [1992] (IGP Berichte Nr. 194 und 195) sind erste Arbeiten beschrieben worden, wie man mit Kollokationsansätzen 3D-Meteomodelle für die Schweiz in GPS-Messungen einführen kann. Im nächsten Schritt wurde die Machbarkeit und Zuverlässigkeit solcher Modelle abgeklärt und beurteilt. Die atmosphärenphysikalischen Grundlagen zur troposphärischen Modellbildung und die verschiedenen Methoden zur Datenerhebung, welche sich im wesentlichen aus ANETZ Messwerten der SMA, lokalen Stationsmeteomessungen, Ballonsondendaten und Wasserdampf-Radiometermessungen zusammensetzen, wurden im IGP-Bericht Nr. 216 [HOEFLINGER, 1993] beschrieben. Besonderes Schwergewicht erhielt dabei die kritische Untersuchung der berechneten Weglängenkorrekturen in Abhängigkeit von Azimut und Elevation.

Im vorliegenden Bericht werden die Entwicklungen zur Wasserdampf-Radiometrie am GGL des IGP dokumentiert. Der Umfang des Sachgebietes macht eine Aufteilung der Dokumentation in 2 Teile erforderlich. Der erste Bericht beschreibt das Verfahren, die entwickelte Hardware sowie die Software für die Steuerung des Messsystems und die Auswertung. In einem zweiten Band sehen wir die Dokumentation der im Rahmen eines EC (NF) Projektes erhaltenen Resultate vor.

Für die sorgfältige Zusammenstellung der Grundlagen, die kritische Beleuchtung der existierenden Lösungsansätze und die speditive Entwicklung der Software für Betrieb, Steuerung und Auswertung sind wir den Herren Dipl.Verm.Ing. M. Cocard und Dipl.-Phys. H. Hirter zu grossem Dank verpflichtet. Ihrem unermüdlichen Einsatz gebührt Lob und Anerkennung. Dr. B. Bürki leitet die Entwicklung des Wasserdampf-Radiometers mit grossem Fachwissen und vielen erfindungsreichen Einfällen. Die Herren Dr. A.Geiger, Dr. B. Neiningen, Dr. H. Richner und Prof. Dr. A. Waldvogel begleiten das Projekt mit kompetenter Beratung und wertvollen Vorschlägen. Der vorliegende Bericht ist zugleich ein Beitrag im Rahmen der IGP-Arbeitsgruppen WISA Nr. 05.25 (Fehlereinflüsse bei Satellitenmessungen) sowie WISA Nr. 05.37 (3D Meteo-Modellierung).

Prof. Dr. H.-G. Kahle  
GGL IGP, März 1994

**Inhaltsverzeichnis**

1. Einleitung.....	1
2. Zum Verfahren der Mikrowellen-Wasserdampf-Radiometrie.....	1
3. Theoretische Grundlagen.....	3
4. Hardwarebeschreibung .....	5
4.1 Hardwarekonzept .....	5
4.2 Mikrowellen-Messteil.....	7
4.2.1 Mechanischer Aufbau des Mikrowellenteils.....	8
4.2.2 Mikrowellen-Elektronik.....	9
4.3 Montierung.....	12
5. Messverfahren.....	19
5.1 Messen der Helligkeitstemperaturen.....	19
5.2 Kalibration der Messungen mittels "Tipping-Kurven" .....	20
6. Die Auswertesoftware.....	23
6.1 Aufbau der WVR Datenbank.....	23
6.1.1 Verknüpfungen der einzelnen Tabellen .....	23
6.2 Anleitung zur WVR-Auswertesoftware.....	25
6.2.1 Einleitung .....	25
6.2.2 Starten der WVR-Datenbank .....	26
6.2.3 Hauptfenster / Hauptmenü .....	26
6.2.4 Einlesen.....	28
6.2.5 Auswerten .....	29
6.2.6 Anzeigen .....	31
6.2.7 Werkzeug .....	35
7.Literatur .....	36
Anhang A.....	38
Anhang B .....	40
Anhang C .....	43
Anhang D.....	49
Anhang E .....	64
Anhang F .....	74

## 1. Einleitung

Der vorliegende Bericht dokumentiert Arbeiten, die im Rahmen des ETH-Forschungsprojekts "Entwicklung von Mikrowellen-Wasserdampf-Radiometern für geodätische Satellitenmessungen mit GPS", Reg. Nr. 08132/41 - 0820.5 durchgeführt wurden. Dieses Projekt ermöglichte unserer Forschungsgruppe die Konstruktion von Hard- und Software, mit der wir einen Beitrag zur Verbesserung der erreichbaren Messgenauigkeit auf dem Gebiet der hochpräzisen GPS-gestützten Positionierung, insbesondere der Höhenkomponente leisten können. Der Umfang des Sachgebiets erfordert eine Aufteilung der Dokumentation in zwei Berichte: Der erste Bericht beschreibt das Verfahren, die entwickelte Hardware sowie die Software für die Steuerung und Auswertung. In einem zweiten Band ist die Dokumentation der im Rahmen mehrerer Projekte erhaltenen Resultate vorgesehen.

## 2. Zum Verfahren der Mikrowellen-Wasserdampf-Radiometrie

Durch das Aufkommen der Satellitengeodäsie und besonders durch das GPS-Verfahren hat die Genauigkeit der geodätischen Messverfahren in den letzten Jahren einen markanten Sprung erfahren. Während bei den klassisch-terrestrischen Messmethoden die Sichtverbindung zwischen zwei zu verbindenden Zielen eine unabdingbare Voraussetzung darstellt, fällt diese beim Einbezug von Satelliten als gemeinsame Fernziele fast gänzlich weg. Damit sind der Wahl von Stationspunkten praktisch keine Restriktionen mehr auferlegt. Die geometrische Satellitengeodäsie gestattete in den Sechzigerjahren erstmals, mittels Richtungsmessungen zu Ballonsatelliten interkontinentale Netzwerke zu errichten<sup>1</sup>. Infolge der erreichten Punktgenauigkeiten von ca. 5 Metern, stellten die so erstellten Netze für geodynamische Untersuchungen aber noch keine befriedigenden Lösungen dar.

Die stark aufgekommene GPS-Messtechnik ermöglichte erstmals die Durchführung von hochgenauen Messungen auf beliebigen Stationspunkten im Feld, ausserhalb der grossen Radioteleskop-Stationen. Diese kostenaufwendigen Spezialinstrumente werden vorwiegend für VLBI Zwecke in der wissenschaftlichen Forschung eingesetzt<sup>2</sup>. Die kleinen und dadurch überall einsetzbaren GPS-Satellitenempfänger liefern 3D-Koordinaten beliebiger Messpunkte im Umkreis von ein paar hundert Kilometern bezüglich eines Zweitempfängers in relativ kurzer Zeit mit Zentimetergenauigkeit. Die Genauigkeiten von solchen differentiellen GPS-Messungen sowie deren rationelle Anwendungsmöglichkeiten werden die grundlegende Umstrukturierung der Navigation, Vermessung und Geodäsie weiter vorantreiben, da sie auch weite Bereiche der Landesvermessung und Geodäsie abzudecken vermögen.

Die Möglichkeiten des GPS im hochpräzisen Modus lassen sich allerdings nur voll ausschöpfen, wenn die systematischen Fehler eliminiert werden. Hierzu gehören Fehler

---

<sup>1</sup> Schmid, H. H. (1972): Konzeptionelle und fehlertheoretische Betrachtungen zur Erstellung eines geodätischen Weltsystems mit Hilfe der Satellitengeodäsie. *Vermessung, Photogrammetrie, Kulturtechnik* 1972, p. 99-124.

<sup>2</sup> VLBI: Very Long Baseline Interferometrie ist ein geodätisches Messverfahren, bei dem mittels Korrelationsrechnung auf zwei Stationen die Laufzeitunterschiede der Rauschstrahlung von Quasaren benutzt wird, um sehr genaue Entfernungen und Erdrotationsänderungen zu bestimmen [Kuehn et al., 1993]

beim Satellitensegment, beim Benutzersegment sowie beim Übertragungsmedium (Troposphäre, Ionosphäre, Multipath). Den **Ionosphäreneinfluss** kann man weitgehend durch Beobachtung auf zwei Frequenzen eliminieren [s. z. B. Wild, 1994], während für die Erfassung des **troposphärischen Effekts** ungleich grössere Anstrengungen unternommen werden müssen.

Neben den optischen Richtungs- und Distanzmessungen sind es vor allem die Mikrowellen-Messungen, bei denen die **Refraktion** eine entscheidende Rolle spielt. Die Refraktion, die zu einer Weglängenvergrößerung (englisch Path length excess oder path delay) führt, ist dabei von zwei Grössen abhängig:

- Der Dichte der trockenen Luft
- Dem Wasserdampfgehalt in der Luft

Somit kann der resultierende totale Path Delay in zwei Anteilen ausgedrückt werden:

$$\Delta s_{\text{total}} = \Delta s_{\text{dry}} + \Delta s_{\text{wet}}$$

$\Delta s_{\text{dry}}$  wird als **Trockenanteil** und  $\Delta s_{\text{wet}}$  als **Feuchtanteil** des gesamten troposphärischen Delays bezeichnet. Obwohl der Trockenanteil ca. 2.3 m erreicht, ist er zuverlässiger modellierbar als der Feuchtanteil, da er in erster Näherung nur vom herrschenden Luftdruck abhängt:

$$\Delta s_{\text{dry}} [\text{cm}] = 0.2277 P [\text{hpa}]$$

Grössere Probleme bietet hingegen der Feuchtanteil, der die Grössenordnung von 1 bis ca. 30 cm erreicht und infolge seiner Natur mit kurzzeitig eintretenden Wetterwechseln stark korreliert ist.

Durch direkte Messung des Wasserdampfgehalts kann der Feuchtanteil des Path Delays (der Einfachheit halber in der Folge als Path Delay bezeichnet) bestimmt, und in Rechnung gesetzt werden. Die Nichtberücksichtigung dieser Systematik wirkt sich vor allem auf die Höhengenaugigkeit der Stationskoordinaten aus, die gegenüber der Lagegenauigkeit um den Faktor 2 bis 3 geringer ist.

Zur messtechnischen Erfassung können passiv arbeitende Mikrowellen Wasserdampf-Radiometer eingesetzt werden, die die von der Troposphäre emittierte Strahlung in zwei abgestimmten Frequenzen messen. Daraus lassen sich der Wasserdampfgehalt und schliesslich die Weglängenverzögerung messtechnisch erfassen [s. z. B. Elgered et al., 1991, Johansson et al., 1993]. Derartige Geräte sind seit geraumer Zeit bei einigen grossen Radioteleskopen im Einsatz, wo mit der VLBI-Technik hochgenaue Basislinien von mehreren Tausend Kilometern Länge für geodätisch/geodynamische Anwendungen bestimmt werden. Diese stationären Radiometer sind aber gross und schwer und eignen sich nicht für den gezielten Einsatz im Feld. Die in diesem Bericht dokumentierten transportablen Radiometer des IGP sind, dank ihres geringen Gewichts, praktisch überall im Feld einsetzbar, was die Messung des troposphärischen Path Delays gezielt am Standort der GPS-Empfänger ermöglicht.

### 3. Theoretische Grundlagen

Das Messprinzip von Mikrowellen-Wasserdampf-Radiometern (WVR) besteht in der passiven Messung der Strahlungsenergie (Helligkeitstemperatur) des Wasserdampfs in zwei Frequenzen; bei **23.8 GHz** und bei **31.5 GHz**. Der Wasserdampf beeinflusst in direktem Mass die Ausbreitungsbedingungen für Signale im Mikrowellenbereich. Er kann die physikalischen Weg der Signale im Bereich von ca. 1 bis 30 cm (bezogen auf den Zenit) verfälschen.

Die **Strahlungs-Transportgleichung** beschreibt die von der Atmosphäre emittierte und am Boden detektierbare Helligkeitsstrahlung:

$$T_B = T_C \cdot e^{-\tau(\infty)} + \int_0^{\infty} T_C \cdot \alpha^{-\tau(s)} ds \quad (1)$$

mit der optischen Tiefe :  $\tau(s) = \int_0^s \alpha ds$

$\alpha$  = Absorptionskoeffizient

$s$  = Distanz entlang dem Ausbreitungsweg  
des Mikrowellensignals

$T$  = Lufttemperatur

$T_B$  = Helligkeitstemperatur

$T_C$  = Kosmische Hintergrundstrahlung ( $\approx 2.8$  K)

Die Absorptionskoeffizienten  $\alpha_i$  der Atmosphäre sind im wesentlichen vom Wasserdampf, dem Sauerstoff und den Wassertropfen (Flüssigwasser in den Wolken) abhängig. Der Wasserdampf hat eine Spektrallinie bei 22.235 GHz. Um nur den Wasserdampfanteil des Path Delays zu bestimmen, wird auf zwei unterschiedlichen Spektrallinien gemessen. Durch eine Linearkombination dieser zwei möglichst optimal gewählten Messungen, kann der Flüssigwasseranteil eliminiert werden (Wu, 1978). Für das WVR des IGP wird das Frequenzpaar 23.8 / 31.5 GHz verwendet .

Für kleine Opazitäten (Opacities) ist die Gleichung (1) linear (Wu et al., 1978). Mit den Näherungen

$$e^{-\tau(\infty)} \cong 1 - \tau(\infty) \quad \text{und}$$

$$\alpha^{-\tau(s)} \cong \alpha$$

kann man die **linearisierte Helligkeitstemperatur  $T'_B$**  vereinfacht schreiben :

$$= T_C + \int_0^{\infty} (T - T_C) \cdot \alpha \cdot ds \quad (2)$$

$$T'_B = T_C \cdot [1 - \tau(\infty)] + \int_0^{\infty} T \cdot \alpha \, ds$$

Eine Lösung der Gleichungen (1) und (2) ist (Wu et al., 1978):

$$T'_B = T_C - (T_{\text{eff}} - T_C) \cdot \ln \left( 1 - \frac{T'_B - T_C}{T_{\text{eff}} - T_C} \right) \quad (3)$$

Für trockene Klimas kann  $T_{\text{eff}}$  mit  $k_e T$  angenähert werden. Typische Werte für  $k_e$  sind  $0,92 < k_e < 0,95$ . Diese Werte können mit Radiosondenaufstiegswerten berechnet werden (Wu et al, 1978). Für die Temperatur  $T$  ist das Tagesmaximum einzusetzen.

Die Berechnung des Path Delays kann nun in ein System mit zwei linearisierten Gleichungen mit zwei Unbekannten überführt werden. Die zwei Unbekannten sind: der Feucht-Wasserdampf-Path Delay und der integrierte Flüssigwasseranteil. Damit erhalten wir für den Feuchtanteil des Path Delays (wet path delay):

$$\Delta L_v(\Phi) = a_0 m + a_1 T'_{B,1} + a_2 T'_{B,2} \quad (4)$$

$$m = \text{Luftmasse} = \frac{1}{\sin(\text{Elevation})}$$

$T'_{B,1}$  = linearisierte Helligkeitstemperatur bei 23,8 GHz  
(Wasserdampf)

$T'_{B,2}$  = linearisierte Helligkeitstemperatur bei 31,5 GHz  
(Flüssigwasseranteil)

Die sog. **Inversionskoeffizienten**  $a_0$ ,  $a_1$ ,  $a_2$  sind vom regionalen statistischen Klima abhängig und werden aus Radiosondierungen berechnet. Für die Radiometer des IGP wurden Datensätze von 1989 verwendet, die uns freundlicherweise von der Schweizerischen Meteorologischen Anstalt (SMA) in Payerne zur Verfügung gestellt wurden. Die Analyse ergab folgende Werte:

$$\begin{aligned} a_0 &= -0.696 \text{ cm} \\ a_1 &= 0.530 \text{ cm/K} \\ a_2 &= -0.302 \text{ cm/K} \end{aligned}$$



## 4. Hardwarebeschreibung

### 4.1 Hardwarekonzept

Mikrowellen-Wasserdampf-Radiometer sind infolge der benötigten Hochfrequenzbauteile sehr kostenintensiv. Die Instrumente, die bisher entwickelt wurden, sind entweder zu teuer und oder zu gross und nur stationär einsetzbar. Bei der Entwicklung der IGP-Radiometer stand daher vor allem die leichte Transportierbarkeit im Vordergrund, im weiteren sollte sich zudem der Stromverbrauch in Grenzen halten.

Als integrierender Bestandteil des Hardwarekonzepts wurde bei der Firma CAPTEC; Consulting, Applied Physics and Software Engineering in Bern, eine Projektstudie<sup>3</sup> in Auftrag gegeben, die die technische Realisierbarkeit des Vorhabens zum Inhalt hatte.

Diese Untersuchung zeigte, dass es möglich ist, ein für geodätische Einsätze geeignetes Gerät zu entwickeln, das nicht nur die geforderte Messgenauigkeit aufweist, sondern auch tragbar ist und damit im Gelände eingesetzt werden kann. In der Folge wurden mit der Firma CAPTEC Gespräche über die konkrete Realisierung des Vorhabens geführt. Auf Grund dieser Vorgaben sowie aus Kostengründen drängte sich eine Arbeitsteilung zwischen CAPTEC und dem IGP auf. Da auf Seiten der Lieferfirma noch keine Erfahrung im Bau computergesteuerter Nachführungen bestand und andererseits dieser Problemkreis im Rahmen diverser Seminar- und Studienarbeiten im Geodäsie- und Geodynamik Labor (GGL) des IGP bereits bearbeitet wurde<sup>4</sup>, erschien folgende Aufteilung als sinnvoll:

1. Die Firma CAPTEC wurde mit dem Bau der Mikrowellen-Radiometer-Messteile sowie den automatischen Meteo- Datenloggern beauftragt und
2. Das IGP besorgte den Entwurf und die Konstruktion der Montierung, mit Einschluss der computergesteuerten motorischen Nachführung und der dazugehörigen Elektronik für die System-Stromversorgung, die Ansteuerung der Schrittmotoren sowie für den Zusammenschluss aller übriger Komponenten und Sensoren mit dem PC als Steuerrechner.

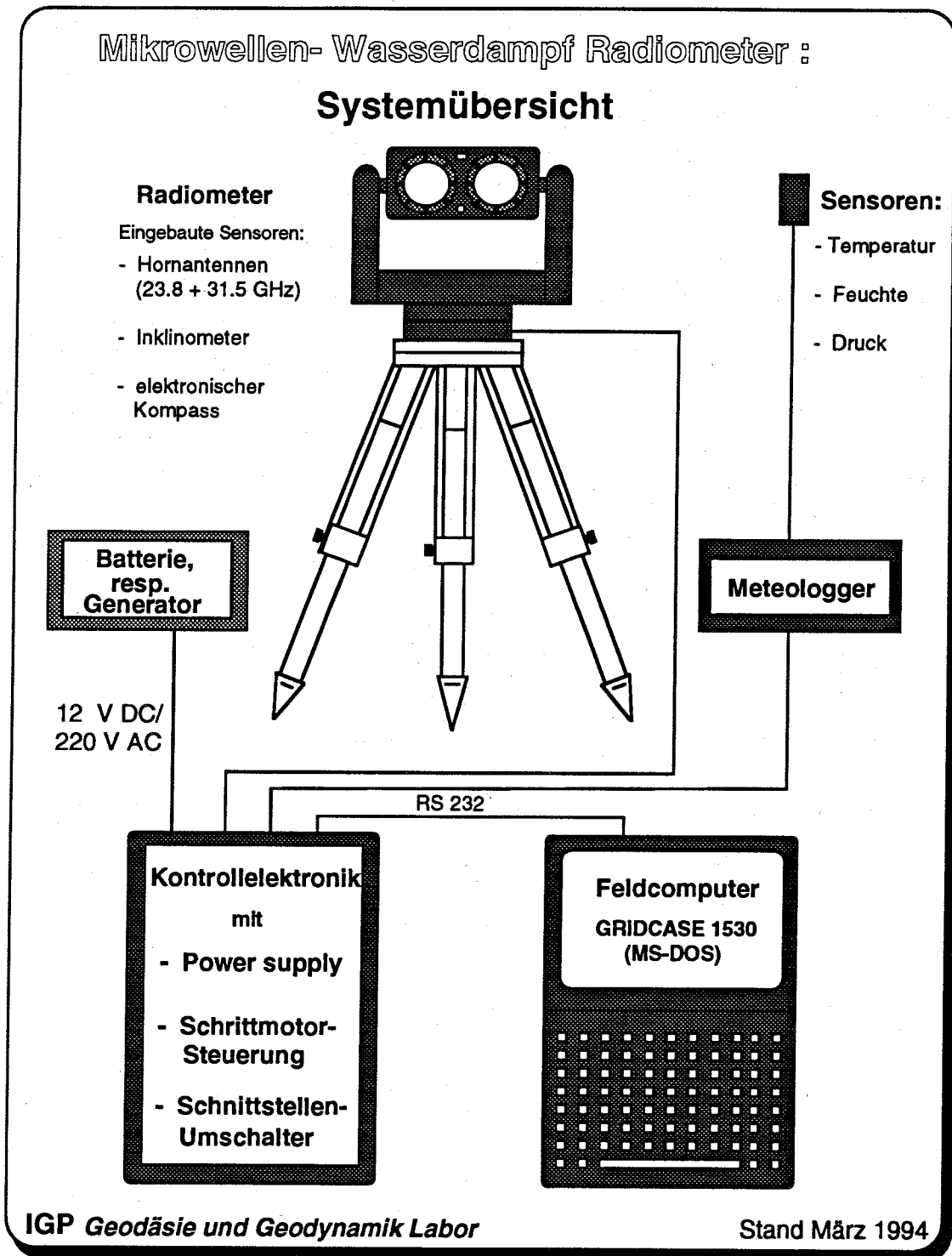
Aus diesen Rahmenbedingungen heraus wurde ein Messsystem entwickelt, das sich folgendermassen charakterisieren lässt:

Das Radiometer, das sich für den Transport in zwei Teile zerlegen lässt, wiegt insgesamt nur 28 kg. Es kann auf einem normalen Vermessungsstativ montiert und mit Netzstrom, einem kleinen Stromgenerator oder einer Autobatterie betrieben werden. Das Gerät kann mittels den in der Montierung integrierten Schrittmotoren und der entsprechenden Steuerelektronik in zwei Achsen bewegt und computergesteuert in jede beliebige Richtung orientiert und nachgeführt werden. Zur zuverlässigen Kontrolle der

<sup>3</sup> Gyger, R., Künzi, K. und Kämpfer, N. (1988): Wasserdampf-Mikrowellen-Radiometer zur Bestimmung der atmosphärischen Weglängenkorrektur. Theorie und praktische Realisierung für die Anwendung in der Geodäsie. Bericht zu Handen des Instituts für Geodäsie und Photogrammetrie der ETH - Zürich.

<sup>4</sup> Oswald, A. (1989): Computergesteuerte Nachführung von astronomischen Kleinteleskopen. Seminararbeit am Institut für Geodäsie und Photogrammetrie.

räumlichen Ausrichtung des Instruments, die zum Beispiel durch starken Winddruck verändert werden kann, sind im Mikrowellenteil ein Flüssigkeits-Inklinometer (zur Messung der Elevation) und in der Montierung ein elektronischer Fluxgate-Kompass zur Erfassung der azimuthalen Ausrichtung integriert. Diese Spezialsensoren werden bei den Messungen vom Computer eingelesen und mit den eigentlichen Messdaten abgespeichert.



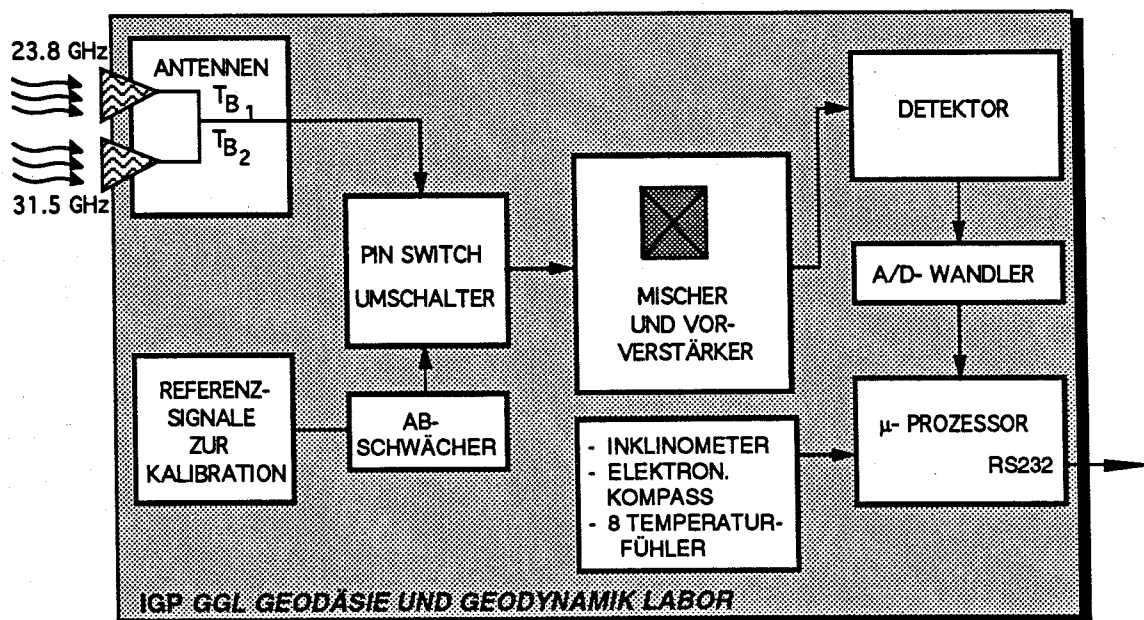
Figur 1: Die Komponenten des Radiometer-Messsystems des IGP-GGL

Für die Erhebung der meteorologischen Stationswerte gehört ein zusätzliches Gerät zur Systemausrüstung. Dieser Meteo- Datenlogger kann auch für andere Zwecke eingesetzt werden, da es über einen eingebauten Mikrocomputer verfügt, der bis zu 2000 Messwerte speichern kann. Durch die computergerecht ausgelegte Ansteuerung sämtlicher Systemkomponenten kann das Radiometer automatisch arbeiten. Da das System bis auf den Steuerrechner und die Meteo-Ausrüstung vollständig in einer Transportkiste Platz findet, können bei gut zugänglichen Stationen sowohl die GPS- wie auch die Radiometermessungen von einer zweiköpfigen Beobachterequipe ohne Probleme durchgeführt werden. Figur 1 zeigt eine schematische Übersicht mit allen Systemkomponenten.

#### 4.2 Mikrowellen-Messteil

Der Radiometer-Mikrowellenmessteil hat die Aufgabe, die Strahlungsintensität, resp. die Helligkeitstemperatur der von der Atmosphäre emittierten Strahlung zu messen. Dazu ist das Radiometer mit zwei passiv arbeitenden Hornantennen ausgerüstet. Vorzugsweise erfolgen diese Messungen in zwei verschiedenen Frequenzen, um die vom Wasserdampf und Flüssigwasser emittierten Anteile separieren zu können. Als geeignet haben sich die Frequenzen von 23.8 GHz und 31.5 GHz erwiesen (Wu, 1979). Das von den Hornantennen empfangene Rauschsignal wird im Mischer vom hochfrequenten Bereich auf eine Zwischenfrequenz von 110 MHz heruntergemischt, um ca. 80 dB verstärkt und auf eine Diode geführt, die das anstehende Signal detektiert und als analoge Spannung anlegt, die der einfallenden Rauschtemperatur proportional ist.

Figur 2 zeigt die prinzipielle Funktionsweise des Radiometers:



Figur 2: Prinzipschema der IGP-GGL Radiometer

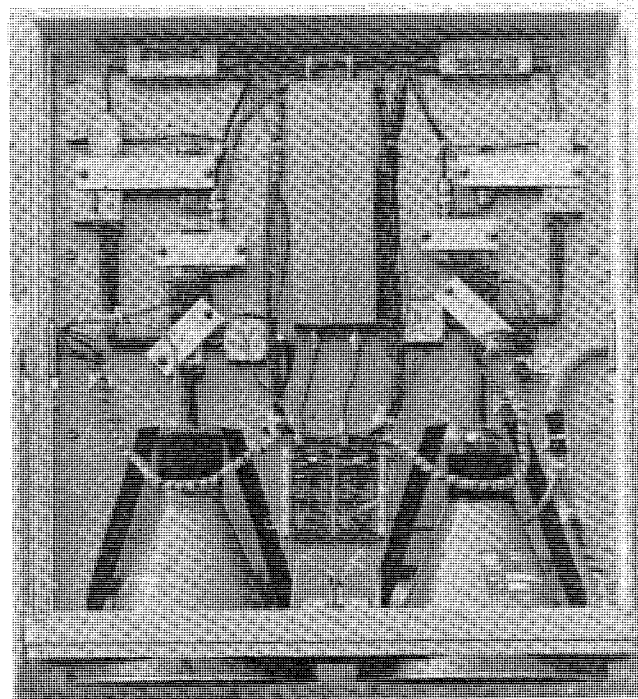
Die Strahlungstemperatur wird aus Vergleichsmessungen mit einer "kalten" und einer "heissen" Kalibrationslast bestimmt. Durch diese stetige interne Kalibration wird ein Wegdriften der Messwerte verhindert. Die Messung der Heisslast wird durch eine Referenz-Rauschdiode realisiert, an die eine Spannung von 28 Volt angelegt wird, während die Kaltlast zur selben Diode, aber ohne Spannung erfolgt, was eine Kaltlasttemperatur ergibt, die der Strahlung eines Schwarzkörpers bei Umgebungstemperatur entspricht.

Für weitergehende technische Angaben wird an dieser Stelle verzichtet und auf die Projektstudie sowie auf die Gesamt-Dokumentation der Instrumente verwiesen.

#### 4.2.1 Mechanischer Aufbau des Mikrowellenteils

Entsprechend den mit der Lieferfirma vereinbarten Konstruktions-Spezifikationen wurde der Mikrowellen-Messteil, das eigentliche Radiometer also, als selbständig zu betreibende Einheit gebaut:

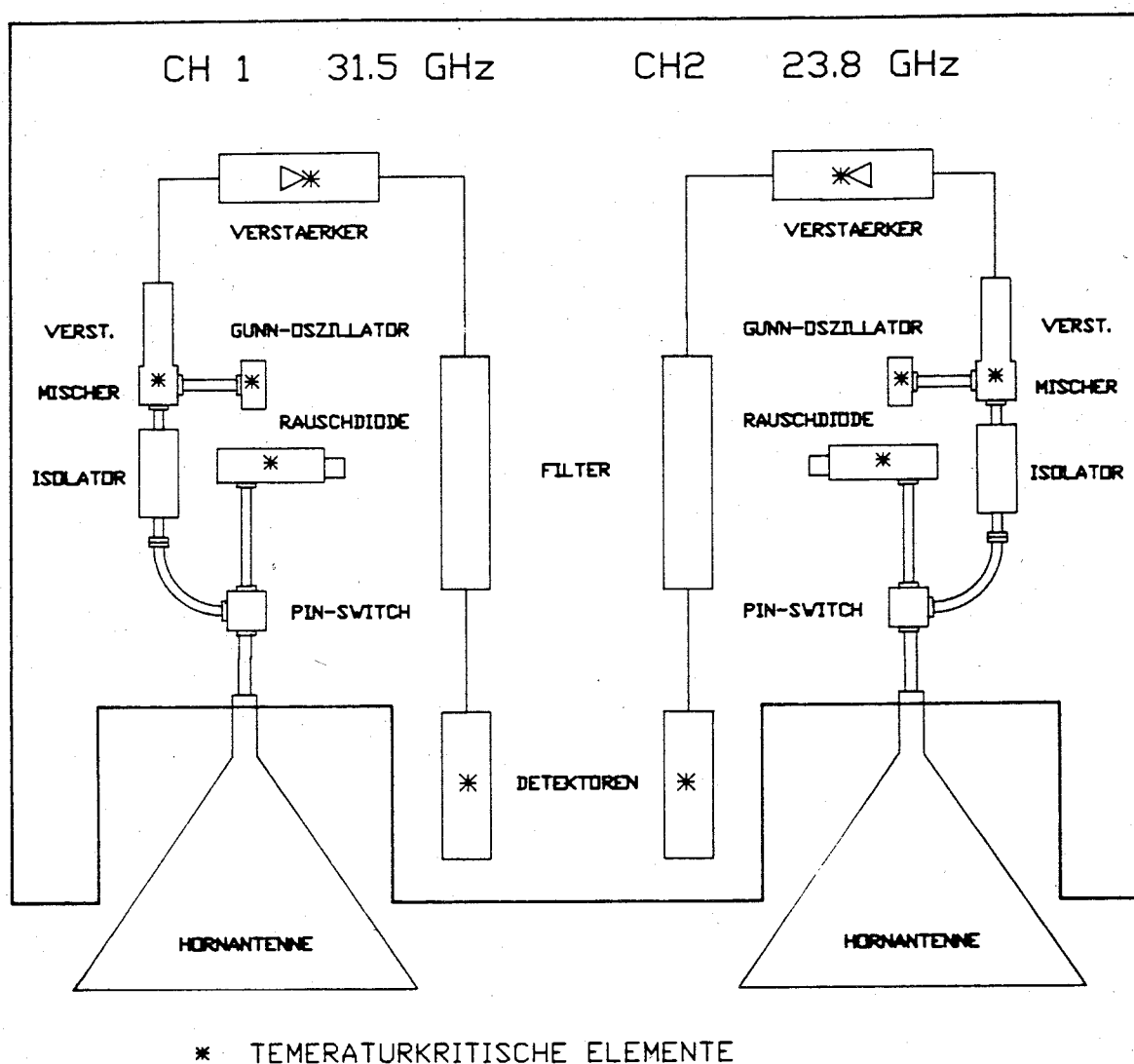
Der eigentliche Messteil ist auf einer 3 mm dicken Aluminium-Grundplatte so aufgebaut, dass auf der einen Seite sämtliche Mikrowellen-Bauteile, und auf der anderen Seite alle elektronischen Schaltkreise übersichtlich und servicefreundlich aufgebaut sind. Auf den beiden Längsseiten ist die Grundplatte von zwei Profilschienen abgeschlossen, die einerseits Führungsnuten für die Aufnahme von zwei thermisch isolierten Gehäuse-Halbschalen aufweisen und andererseits mit den mechanischen Achszapfen für die Einhängung in die Montierung versehen sind. Figur 3 zeigt die (für die Aufnahme vertikal ausgerichtete) Grundplatte mit den darauf montierten Mikrowellenteilen:



Figur 3: Aufbau der Mikrowellen-Bauteile auf der Grundplatte des Radiometers.

Die beiden am unteren Bildrand sichtbaren kegelförmigen Hornantennen sind auf ein in rechtwinkliger Anordnung zur Grundplatte montiertes Frontblech geschraubt. Sie sind mechanisch nur indirekt über die Hohlleiter mit der Grundplatte verbunden. Zur Kontrolle des thermischen Verhaltens sind an den temperaturkritischen Bauteilen Sensoren zur Messung der Temperatur angebracht

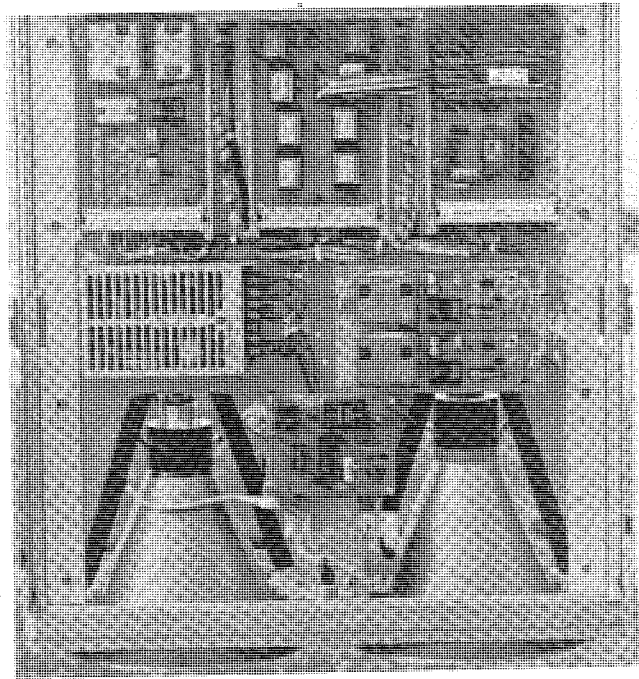
Figur 4 zeigt ein Blockschaltbild mit den Mikrowellen-Komponenten:



Figur 4: Blockschaltbild der Mikrowellen-Bauteile auf der Grundplatte des Radiometer-Messteils.

#### 4.2.2 Mikrowellen-Elektronik

Die für den Betrieb des Mikrowellenteils benötigte Elektronik ist auf der Rückseite der Grundplatte in fünf Baugruppen montiert:



Figur 5: Anordnung der Elektronik-Baugruppen im Radiometer.

Die einzelnen Platinen oder Module dienen folgenden Aufgaben:

- **Platine 1** (in Fig. 5 rechts oben plaziert) enthält den Mikrocomputer Tattletale IV als lokale Intelligenz zur Steuerung des Radiometers sowie einige externe Bauteile. Dieser Einplatinencomputer weist Dimensionen von 6 cm x 9.5 cm auf. Er verfügt über zwei nichtflüchtige Speichersegmente, die für das Betriebsprogramm sowie für die Messdaten zur Verfügung stehen. Zusätzlich befinden sich noch ein 12-Bit Analog-Digital Wandler, ein Multiplexer Input/output Board, sowie die RS232-Schnittstellentreiber auf der Platine. Das Betriebsprogramm kann vom PC aus über die serielle Schnittstelle direkt in den Programmspeicher geladen werden.

- **Platine 2** (obere Reihe links) enthält verschiedene DC/DC-Converter, die aus der Batteriespannung folgende Betriebsspannungen generieren:

-12 V DC / 100 mA  
+28V DC / 100 mA  
+ 5 V DC / 900 mA

Daneben ist die Platine mit Relais zur Zuschaltung der Rauschdioden und der PIN-Schalter bestückt.

- **Platine 3** (obere Reihe, Mitte) dient der Versorgung, Erfassung, Umformung und Digitalisierung der Temperatur-Messfühler, von denen insgesamt acht auf den temperaturkritischen Bauteilen montiert sind. Die Messwerte werden parallel zu den Analogspannungen des Detektors gemessen und erlauben die Kontrolle des thermischen Verhaltens der Bauteile.

- **Platine 4** (untere Reihe, links) ist mit DC/DC- Wandlern bestückt, die die Betriebsspannungen für die Gunn-Oszillatoren erzeugen. Gunn-Oszillatoren reagieren empfindlich auf Überspannungen, deshalb sind in der Elektronik sog. Crowbar-Schaltkreise integriert worden. Diese arbeiten mit Thyristoren, die die Ausgangsspannungen absenken. Die Thyristoren sind im weiteren mit Schmelzsicherungen vor Überhitzung abgesichert.

- **Platine 5** (untere Reihe, links) ist eine Baugruppe mit der ersten Filter- und Glättungsstufe Versorgungsspannung für sämtliche Bauteile. Sie regelt die von der angeschlossenen Versorgungselektronik, einer Autobatterie oder einem Generator eingespiesene unregelmäßige Gleichspannung, die im Bereich 12 bis 15 Volt liegen kann.

Zusätzlich zum Betrieb im Verbund mit der motorgetriebenen Montierung kann das Radiometer auch unabhängig und ohne direkte Verbindung zur Steuerelektronik in Betrieb genommen werden. Dazu dient ein kleines, als RCU (Remote Control Unit) bezeichnetes Steuergerät, das über ein Kabel mit dem Radiometer verbunden wird, und über einen Anschluss für die serielle Schnittstelle zum PC verfügt. Diese Betriebsmöglichkeit ist vor allem für Kontroll- und Kalibrationsmessungen, zur Fehlersuche sowie für Langzeit-Laborversuche vorgesehen, bei denen keine Nachführung benötigt wird.

### 4.3 Montierung

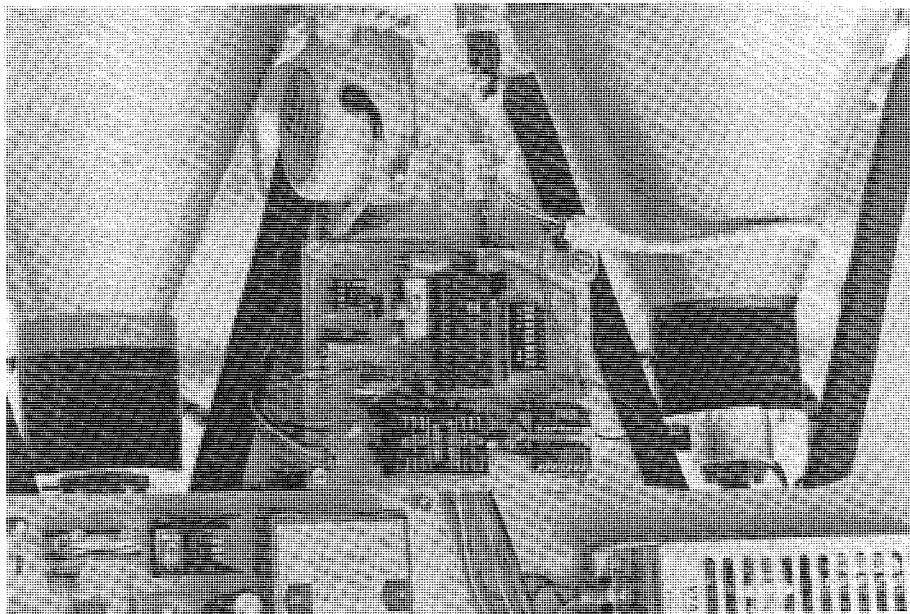
Entsprechend der vereinbarten Arbeitsteilung fielen Entwurf und Konstruktion der Gerätemontierung sowie der übrigen Komponenten in den Aufgabenbereich des IGP. Zu Beginn der Konstruktionsarbeiten musste zunächst die Montierungs- und Antriebsart festgelegt werden. Da das Radiometer im topozentrischen Koordinatensystem (mit Azimut- und Elevationsachse) nachzuführen ist, eignet sich am besten eine azimutale Gabelmontierung. Im Gegensatz zu anderen, wie z. B. der parallaktischen, bietet die azimutale Montierung durch ihre konstruktive Symmetrie keine Probleme mit der Befestigung des Instruments und erfordert ausserdem keine Gegengewichte, was die zu bewegende Masse nicht unnötig vergrössert.

Bei den konzeptionellen Überlegungen zur Antriebsart mussten Vor- und Nachteile der in Frage kommenden Motortypen abgeklärt werden. Für ruckfreie und fein arbeitende Nachführungen eignen sich grundsätzlich Gleichstrom-Servomotoren sehr gut. Im Vergleich dazu ergeben Schrittmotoren die etwas "gröbere" Antriebsart, vor allem im Bereich der Resonanzgeschwindigkeit(en). Unter der Voraussetzung, dass keine Schritte verloren gehen, und der jeweils aktuelle Zählerstand bekannt ist, weisen die Schrittmotoren den grossen Vorteil auf, dass man stets genau weiss, in welche Richtung die angetriebene Achse ausgerichtet ist. Zudem bleiben die Gleichlaufschwankungen minimal. Im Gegensatz dazu muss bei der Ansteuerung mit Servomotoren die Position der Achse ständig gemessen und im Regelkreis rückgeführt werden. Dazu sind normalerweise auf der angetriebenen Achse oder auf dem Motorgehäuse Drehgeber montiert, die pro Achsumdrehung eine bestimmte Anzahl Pulse generieren, die zur Positionskontrolle ständig detektiert und gezählt werden müssen. Für den praktischen Betrieb erfordert dies Regelkreise, die einerseits die Motoren mit der benötigten Spannung versorgen und andererseits eine ständige Positions- und Geschwindigkeitskontrolle durchführen. Während für viele Laboranwendungen Systemlösungen für solche Steuerungen mittels PC auf dem Markt erhältlich sind, ist deren Anwendung unter Feldbedingungen aufwendig, problematisch und erfordert vor allem grossen Software-Aufwand, hauptsächlich im Zusammenhang mit selbst entwickelter Betriebssoftware wie für das Radiometer. Ein weiterer Punkt besteht in der geforderten Positionierungsgenauigkeit, die bei Radiometermessungen eine eher untergeordnete Rolle spielt, was deshalb eher für den Schrittmotor spricht. Diese Überlegungen führten schliesslich zum Entschluss, die Radiometer-Montierung mit kräftigen Schrittmotoren anzutreiben, zumal auch das in unserem Labor eingesetzte Kleinteleskop mit dieser Antriebsart ausgestattet ist, und wir damit gute Erfahrungen sammeln konnten. Im Hinblick auf spätere Anwendungen wurde bei der seinerzeitigen Entwicklung der Steuerelektronik für das Kleinteleskop bereits darauf geachtet, dass auch Schrittmotoren mit höherer Leistung angetrieben werden können, was lediglich eine andere Bestückung der Endstufe bedingt.

Die im Laufe der praktischen Feldeinsätze gewonnenen Erfahrungen mit dem Schrittmotorantrieb haben gezeigt, dass diese Antriebsart bei ungestörten Wetterverhältnissen durchaus geeignet ist. Starke Windböen können aber die Orientierung des Instruments verändern, was von der Elektronik nicht erkannt werden kann, da bei Schrittmotoren in der Regel eine Rückkopplung fehlt, respektive nicht notwendig ist. Diese Erkenntnisse führten zum Entschluss, zusätzlich geeignete Sensoren zur Kontrolle der Orientierung einzubauen. Die **Elevation** des Radiometers kann nun mittels eines Flüssigkeits-Inklinometers gemessen werden (Figur 6). Dies ermöglicht eine stetige Kontrolle und allfällige Korrektur der Motorsteuerung. Zudem kann man damit einer

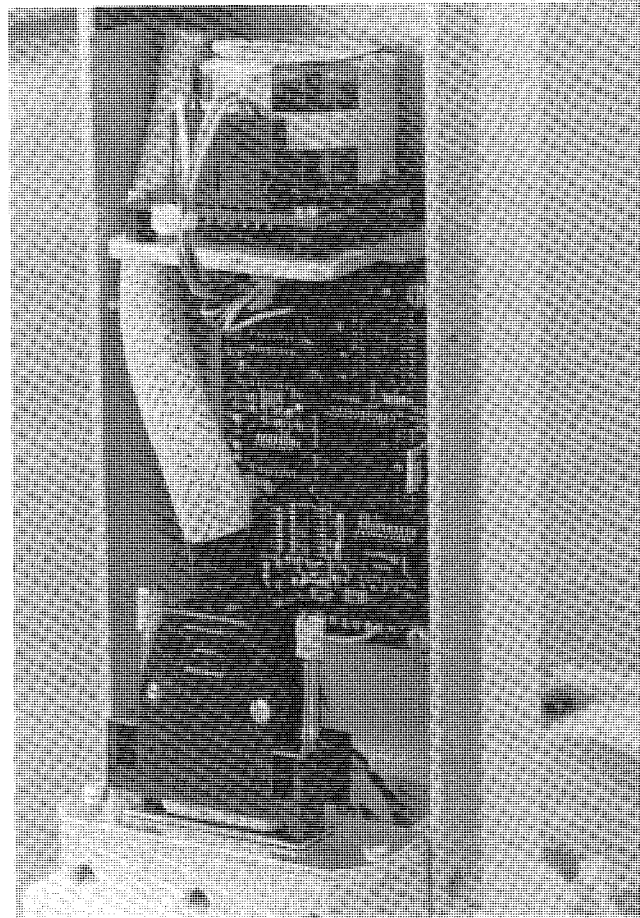


eventuellen Ungenauigkeit der Schrittmotor-Positionierung entgegenwirken, da die effektive Elevation während der radiometrischen Messung erfasst und mit den übrigen Messwerten zusammen abgespeichert wird.



*Figur 6: Anordnung des Flüssigkeits-Inklinometers innerhalb des Mikrowellen-Messteils. Auf der kleinen Zusatzplatine sind die bipolare Stromversorgung, der A/D-Wandler sowie die Treiber für die serielle Schnittstelle RS 232 sichtbar.*

Für die **azimutale Ausrichtung** der Instrumente wurden Flüssigkeitskompass (inklusive der dazugehörigen Versorgungs- und Auswerteelektronik) in die Stützen der Montierung integriert und sind direkt vom PC aus über die serielle Schnittstelle ansteuerbar. Sie liefern das Azimut der Montierung mit einer Genauigkeit von ca. 0.5 Grad, was für die Radiometeranwendung durchaus angemessen ist. Diese neuartigen Sensoren sind klein (4 x 4 x 2.5 cm) und bestehen aus einem Lexan-Gehäuse, in dem ein magnetischer Ringkern (toroidal fluxgate sensing element) in einer inerten Flüssigkeit schwimmt. Dieser Ring wird mittels Primärwicklungen, die das Gehäuse umgeben, elektrisch induktiv gesättigt und richtet sich in Richtung des Erdmagnetfeldes aus. Die geographische Ausrichtung bezüglich einer Referenzrichtung wird durch die Sekundärwicklungen gemessen, ausgewertet, digitalisiert und zur seriellen Schnittstelle geleitet. Die komplette, mit einem speziellem Mikroprozessor bestückte Elektronik, ist senkrecht unterhalb dem Sensorgehäuse angebracht (s. Figur 7). Zur Berücksichtigung lokaler magnetischer Anomalien und Störungen kann der Kompass via PC geeicht werden. Die entsprechenden Daten werden dabei automatisch erfasst und in einem nichtflüchtigen EEPROM abgelegt. Im weiteren können die Integrations- und Dämpfungszeit sowie weitere Parameter programmiert werden.

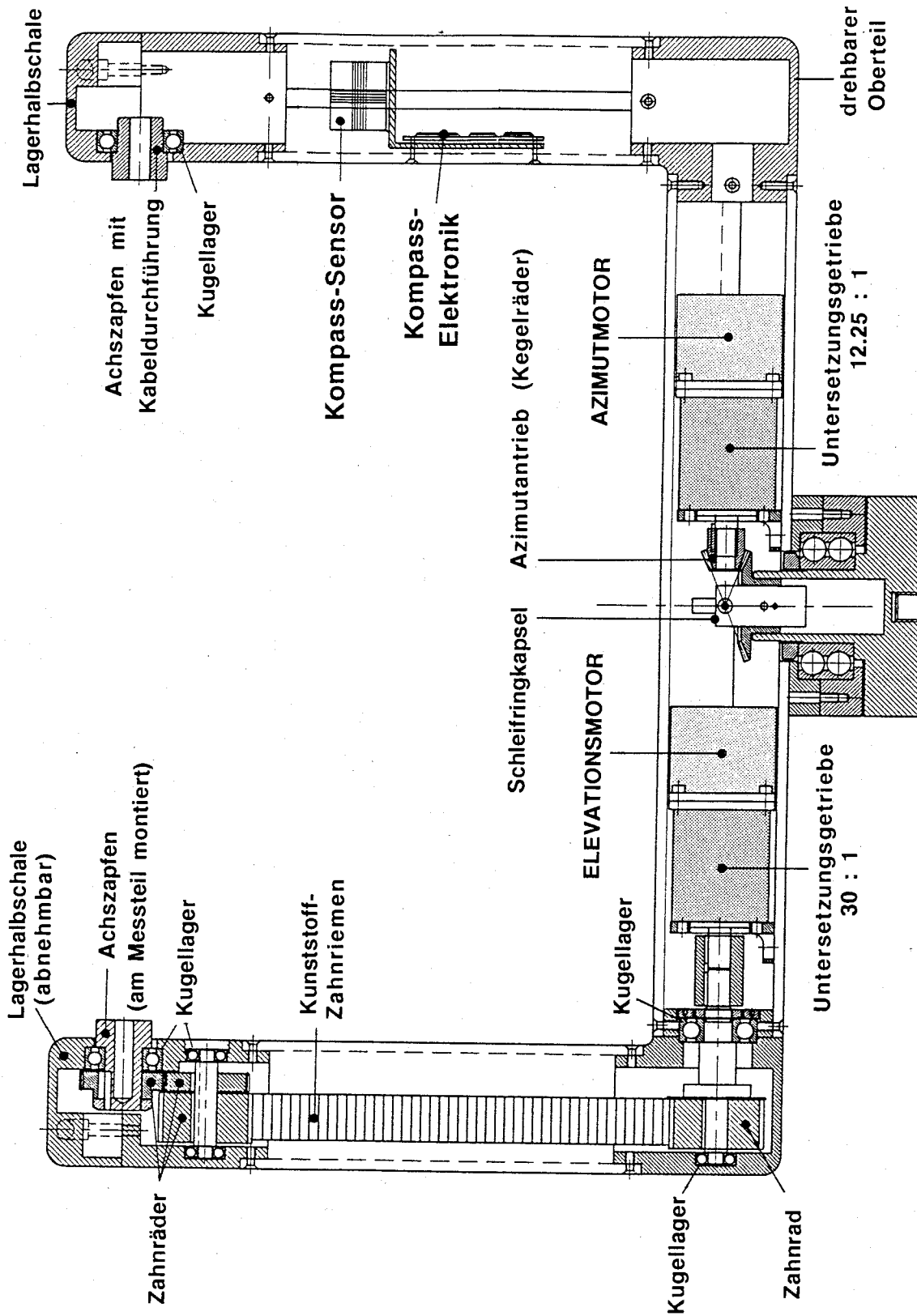


*Figur 7: In der Radiometer-Stütze eingebauter elektronischer Flux-gate-Kompass. Unterhalb des horizontal montierten Sensors ist die mit einem spez.  $\mu P$  ausgerüstete Elektronik fixiert.*

Ein wichtiger Vorteil dieser "absolut" arbeitenden Sensoren besteht in der Möglichkeit, die räumliche Ausrichtung des Radiometers nach einem Ausfall oder Unterbruch der Versorgungsspannung im Zuge eines automatischen Neustarts abzufragen und die Messreihe ohne manuelle Neuorientierung fortzusetzen .

Eine wesentliche Randbedingung für den **mechanischen Entwurf** der Montierung war der Betrieb auf einem normalen Stativ für Vermessungsinstrumente. Das Gewicht des ganzen Gerätes (der Mikrowellenteil wiegt rund 15 kg) sollte 30 kg nicht überschreiten, was letztlich mit einem Gesamtgewicht von 28 kg eingehalten werden konnte. Der Entwurf der Montierung erfolgte durch den Konstrukteur des IGP, Ing. HTL H.P. Oesch, der dazu erstmals vom neuen CAD- System des Instituts Gebrauch machen konnte. Aufgrund der mit diesem System erstellten Werkzeichnungen konnten sämtliche Werkstücke am Institut für Werkzeugmaschinenbau und in den Zentralwerkstätten der KIL-Betriebe angefertigt werden. Dank dieser guten Zusammenarbeit konnte eine kostensteigernde auswärtige Vergabe von Arbeiten vermieden werden.

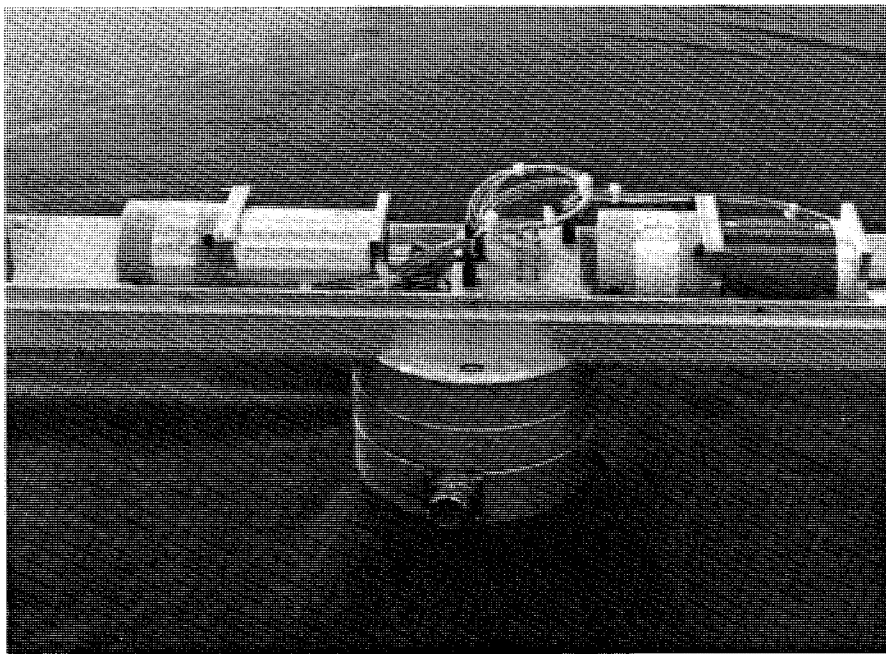
Figur 8 zeigt einen Querschnitt der in zwei Ausführungen angefertigten Montierung:



Figur 8: Querschnitt durch die Gabelmontierung der GGL-IGP- Radiometer. Der Mikrowellenmessteil ist in der Zeichnung nicht eingetragen.

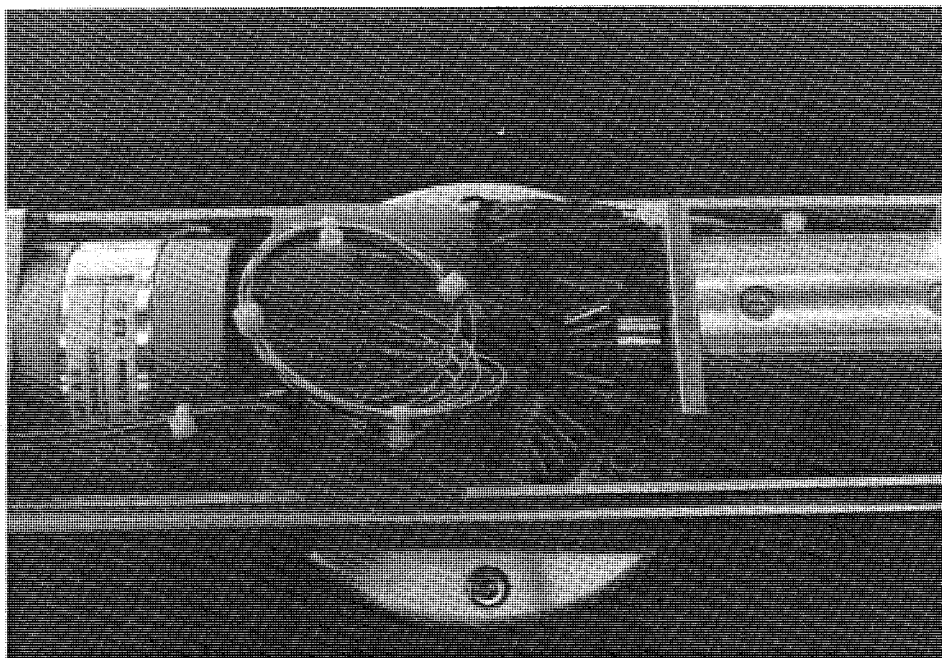
Die Montierung weist folgende technische Einzelheiten auf:

- Der Querschnitt der Gabelmontierung besteht aus zwei U-förmigen Profil-Halbschalen, die sich zu einem Rechteckprofil zusammenschrauben lassen. Die elektrischen Anschlüsse, Antriebsmotoren, Untersetzungsgetriebe und der elektronische Kompass sind auf der unteren, respektive inneren Halbschale montiert. Dadurch kann für Wartungs-, Mess- oder Kontrollzwecke die zweite Halbschale entfernt werden.
- Die oberen Lager-Halbschalen, die den Abschluss der vertikalen Stützen bilden, können gelöst, und der Radiometer-Messteil mit den Achszapfen in die Lager eingehängt werden. Die elektrische Verbindung für die Stromversorgung und die Verbindung zum PC erfolgt durch ein Kabel, das durch das Kugellager des Achszapfens zu einem innerhalb der Stütze montierten Steckeranschluss geführt wird.
- Der motorische Antrieb für die computergesteuerte Nachführung ist vollständig in der Montierung integriert und wird von zwei 4-Phasen bipolar-Schrittmotoren mit 200 Schritten ( $1.8^\circ$ ) pro Umdrehung und einer Stromaufnahme von max. 2 A pro Phase bewerkstelligt:



*Figur 9: Geöffnete Montierung mit den beiden Schrittmotoren für die Nachführung. Links sind der Motor und das Untersetzungsgetriebe für den Azimutantrieb, rechts die Antriebseinheit für die Elevationsachse sichtbar. Die Zuführung der Stromversorgung und der Computersignale erfolgt über ein dreiteiliges Speziakabel zum Unterteil der Montierung (Kupplungen sind unten sichtbar) und wird mittels einer 24-poligen Schleifringkapsel zum drehbaren Teil der Montierung übertragen.*

- Die Azimutachse wird über ein Kegelradgetriebe mit dem Untersetzungsverhältnis 5:2 angetrieben. Das kleinere Kegelrad ist direkt an der Abtriebsachse des Untersetzungsgetriebes des Azimutmotors (Untersetzungsverhältnis 12.25: 1) befestigt. Damit ergibt sich eine Gesamtuntersetzung von 30.625 : 1, was einem Einzelschritt (im Vollschrittbetrieb kleinste ausführbare Richtungsänderung) von 3 Bogenminuten entspricht (Figur 10).



*Figur 10: Kegelradgetriebe des Azimutmotors (Untersetzung 5:2). Das am Unterteil der Montierung festgeschraubte Kegelrad wird durch eine kleine Printplatte halb abgedeckt, auf der die Anschlussdrähte der Schleifringkapsel verlötet sind. Links und rechts davon sind die beiden Antriebseinheiten sichtbar.*

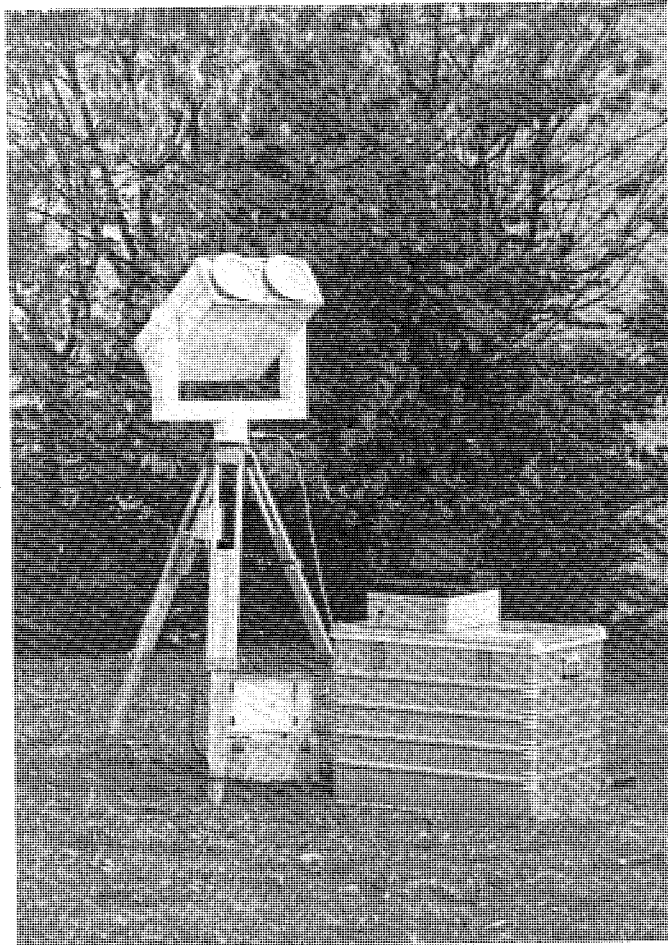
- Die Leistung des Elevationsmotors wird über ein 30:1- Untersetzungsgetriebe auf einen Kunststoff-Zahnriemen übertragen, der den Radiometer-Messteil mittels zwei weiteren Zahnrädern, über einen der beiden seitlichen Achszapfen direkt antreibt. Da diese Achse mit 30:1 untersetzt ist, entspricht ein Einzelschritt wie beim Azimutantrieb einer Richtungsänderung von 3 Bogenminuten.

- Aus wärme- und störungstechnischen Gründen ist die Endstufe der Schrittmotorsteuerung zusammen mit der Stromversorgung und der Logik zur Schnittstellenumschaltung im separaten Elektronik-Gehäuse integriert.

- Die Signale werden in der Azimut-Drehachse über eine im Zentrum des Kegelradgetriebes eingesetzte, 24-polige Schleifringkapsel zum Drehteil übertragen. Bei Instrumentendrehungen entstehen damit keine Probleme mit möglichen Kabelverwicklungen, bzw. -Abrissen.

- Die Steuerung erlaubt die gegenseitig unabhängige Ansteuerung beider Achsen. Die benötigte Zeit für die Ausrichtung des Radiometers in eine beliebige Richtung dauert maximal ca. 6 Sekunden, für die eigentliche radiometrische Messung behält das Instrument die Orientierung für ca. 25 Sekunden bei.

Das komplette Radiometer-Messsystem des GGL-IGP in messbereiter Aufstellung präsentiert sich folgendermassen:



*Figur 11: Das GGL-IGP Wasserdampf-Radiometer in messbereiter Aufstellung.*

## 5. Messverfahren

### 5.1 Messen der Helligkeitstemperaturen

Mit dem Radiometer werden pro Kanal drei verschiedene Quellen angemessen, zum Detektor geleitet, als analoge Spannungen gemessen und zum Mikroprozessor übermittelt. Neben dem Signal des Himmels (sky brightness temperature) werden zwei Referenzsignale erzeugt und gemessen. Als Quelle zur Erzeugung dieser Signale dient eine Rauschdiode, die in zwei verschiedenen Zuständen in das Radiometer eingespiesen und angemessen wird. Im ersten Zustand wird keine Spannung an die Diode gelegt; die Messung bezieht sich demzufolge auf die physische Temperatur eines Schwarzkörpers bei der gemessenen Temperatur im Innern des Geräts (Diese Temperatur wird als sog. **Kaltlasttemperatur** bezeichnet, sie liegt im Bereich von ca. 300 K). Im zweiten Zustand wird an die Diode eine Spannung von 28 Volt angelegt, was bewirkt, dass die Diode mit einer äquivalenten Temperatur von mehreren Tausend Grad "rauscht". Dieses als **Heisslasttemperatur** bezeichnete Rauschsignal wird abgeschwächt und durch die Zwischenfrequenzstufe zum Detektor geführt. Die Umschaltung der Signalquellen (Sky-, Kalt- und Heisslast) auf den Detektor erfolgt dabei durch einen elektronisch gesteuerten Pin-Schalter (pin switch). Zwischen den zwei Referenztemperaturen, der Helligkeitstemperatur der Atmosphäre und den gemessenen Spannungen im Wasserdampfradiometer (WVR) besteht der folgende Zusammenhang:

$$\frac{T_H - T_A}{T_B - T_A} = \frac{V_H - V_A}{V_B - V_A} \equiv \gamma$$

$T_A = T_{\text{Cold}} =$  Mittelwert aller internen Temperaturen pro Kanal des WVR  
(innere Umgebungstemperatur)

$T_H = T_{\text{Hot}} =$  Heisslasttemperatur

$V_A = V_{\text{Cold}} =$  Kaltlastspannung

$V_H = V_{\text{Hot}} =$  Heisslastspannung

$V_B = V_{\text{Sky}} =$  Atmosphärische Helligkeitstemperatur

Gleichung (5) gilt nur, falls keine Verluste und Reflexionen im Horn, Hohlleiter und dem Schaltungssystem entstehen. In der Praxis sind diese Verluste aber nicht ganz eliminierbar. Deshalb muss an der Heisslast eine Korrektur angebracht werden (Wu et al., 1978). Diese **Heisslastkorrektur**  $\Delta T_H$  lässt sich mittels der im folgenden beschriebenen "Tipping-Kurven-Technik" bestimmen.

$$T_B = T_A + (T_H + \Delta T_H - T_A) \cdot \gamma \quad (6)$$

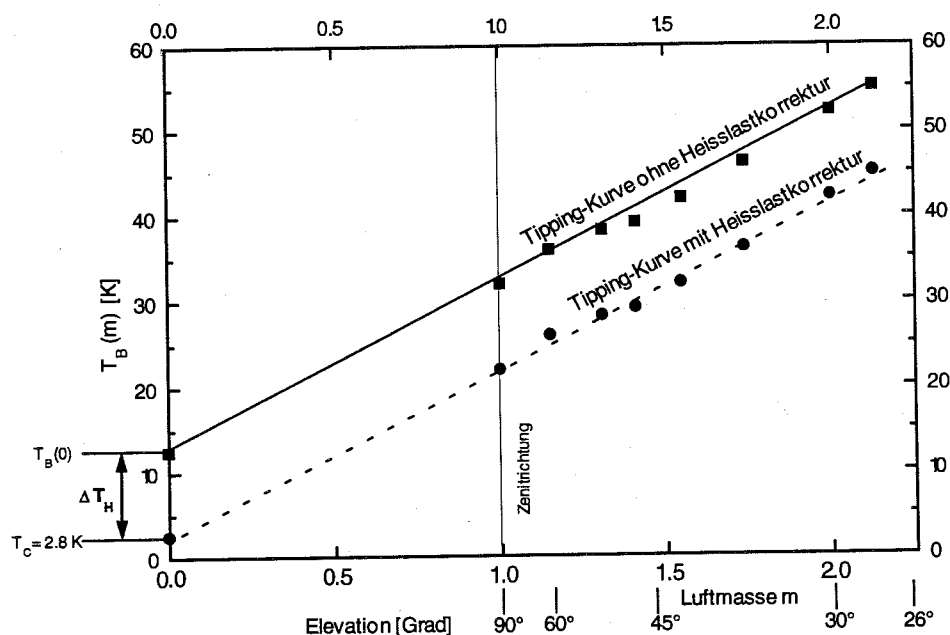
## 5.2 Kalibration der Messungen mittels "Tipping-Kurven"

Die Methode der Tipping-Kurven geht von der Überlegung aus, dass die gemessene Helligkeitstemperatur gemäss Gl. (6) als lineare Funktion der mit dem Messstrahl durchdrungenen Luftmassen dargestellt werden kann (Elgered, 1993). Im Zenit (Elevation 90 Grad) wird durch die Luftmasse  $m=1$ , in 30 Grad Elevation durch die Luftmasse  $m=2$ , etc. gemessen. Zwischen der gemessenen Helligkeitstemperatur und der Luftmasse besteht ein linearer Zusammenhang (Wu et al., 1978). Demzufolge müsste bei einer hypothetischen Messung ohne atmosphärische Luftschichten ( $m=0$  !) ein Wert gemessen werden, der dem kosmischen Hintergrundrauschen von ca. 2.8 K ausserhalb der Erdatmosphäre entspricht. Deshalb kann man durch die Messwerte eine Regressionsgerade berechnen, diese Gerade bis zur Luftmasse  $m=0$  extrapolieren und den Achsenabschnitt  $T_{B(0)}$  berechnen. Der erhaltene Achsenabschnitt wird auf den "Sollwert" von 2.8 K gezwängt. In der Folge sind die Messungen iterativ zu korrigieren, bis die Regression für  $T_{B(0)}$  den Sollwert von 2.8 K ergibt. Diese Korrektur wird als **Heisslastkorrektur**  $\Delta T_H$  bezeichnet (s. Fig. 12)..

Die Bestimmung der Heisslastkorrektur erfolgt durch Messung in acht unterschiedlichen Elevationen pro Tippkurve :

28°, 30°, 35°, 40°, 45°, 50°, 60° und 90°, was den Luftmassen  
 | | | | | | | |  
 2.13, 2.0, 1.74, 1.55, 1.41, 1.31, 1.15 und 1.0 entspricht

Die Tipp-Kurven werden in den Azimuten 0°, 45°, 90°, 135°, 180°, 225°, 270° und 315° gemessen. Die "troposphärische Halb-Sphäre" kann damit innerhalb von ca. 12 Minuten gescannt und erfasst werden.



Figur 12: Tipping-Kurve vor und nach Einbezug der Heisslastkorrektur  $\Delta T_H$



Formel (7) ist eine gute Schätzung für die benötigte Heisslastkorrektur:

$$\Delta T_H = \frac{(T_C - T_{B(0)}) (T_H - T_A)}{(T_{B(0)} - T_A)} \quad (7)$$

Daraus ergibt sich für die Korrektur der linearisierten Helligkeitstemperatur (Wu et al, 1978):

$$\Delta T_B = \Delta T_H \cdot \left( \frac{T_B - T_A}{T_H - T_A} \right) \quad (8)$$

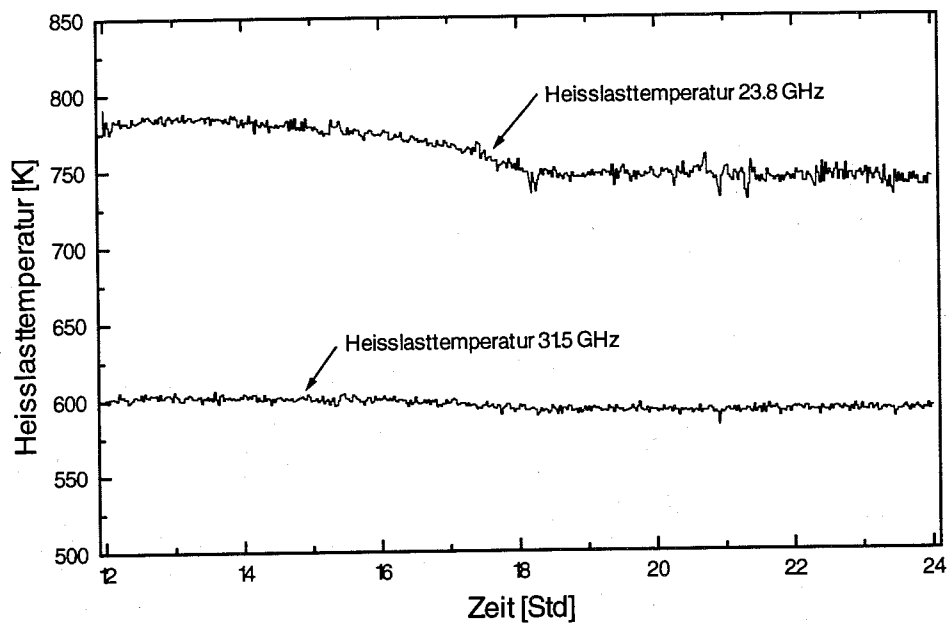
Pro Datensatz, d.h. pro Tipping Kurve wird eine lineare Regression mit einem Startwert für  $T_H$  (ca. 1000 K) durchgeführt. Es wird die Gerade  $T_{Bi} = T_{B(0)} + b m_i$  berechnet.

$$b = \frac{\sum_i m_i T_{B_i} - \frac{\left( \sum_i m_i \right) \left( \sum_i T_{B_i} \right)}{n}}{\sum_i (m_i)^2 - \frac{\left( \sum_i m_i \right)^2}{n}} \quad (9)$$

$$T_{B(0)} = \frac{\sum_i T_{B_i} - b \cdot \sum_i m_i}{n} \quad (10)$$

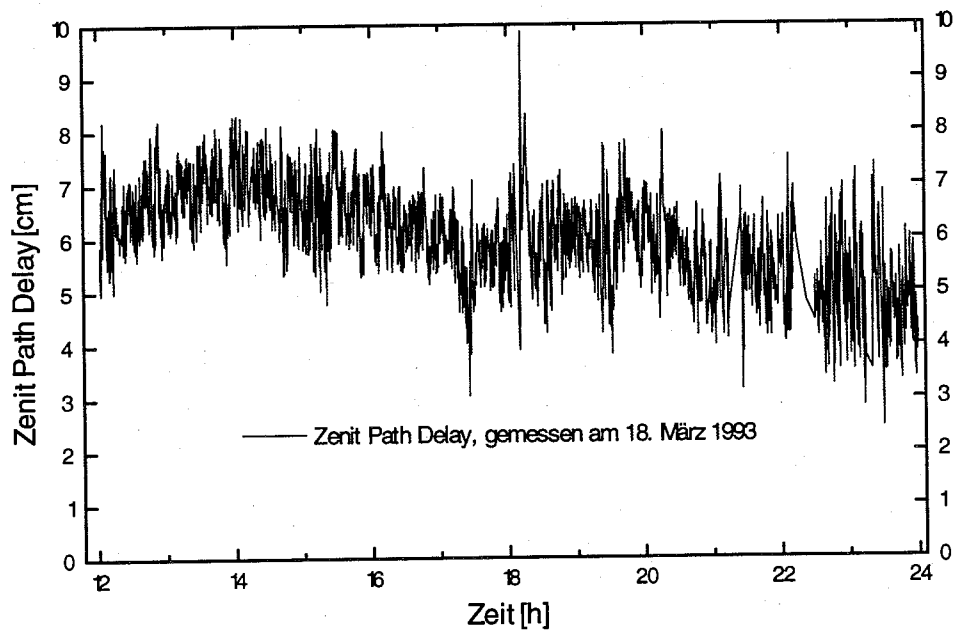
$$\sigma = \sqrt{\frac{\sum_i (T_{B_i})^2 - \frac{\left( \sum_i T_{B_i} \right)^2}{n}}{n-1}} \quad (11)$$

Werden Tipping Kurven über längere Zeit gemessen, so entsteht eine kontinuierliche Messreihe mit Heisslasttemperaturen wie Figur 13 zeigt:



Figur 13: Verlauf der mit Tippkurvenanalyse ermittelten Heisslasttemperaturen für beide Messkanäle in einer Messreihe über 12 Stunden.

Mit den gemessenen Helligkeitstemperaturen der beiden Kanäle kann sodann der Path Delay, in Zenitrichtung (äquivalenter Zenit-Path Delay) unter Zuhilfenahme der Formeln (3) und (4) berechnet werden:



Figur 14: Verlauf des zenitalen Path-Delays, bezogen auf die gleiche Messreihe wie in Fig. 13.

## 6. Die Auswertesoftware

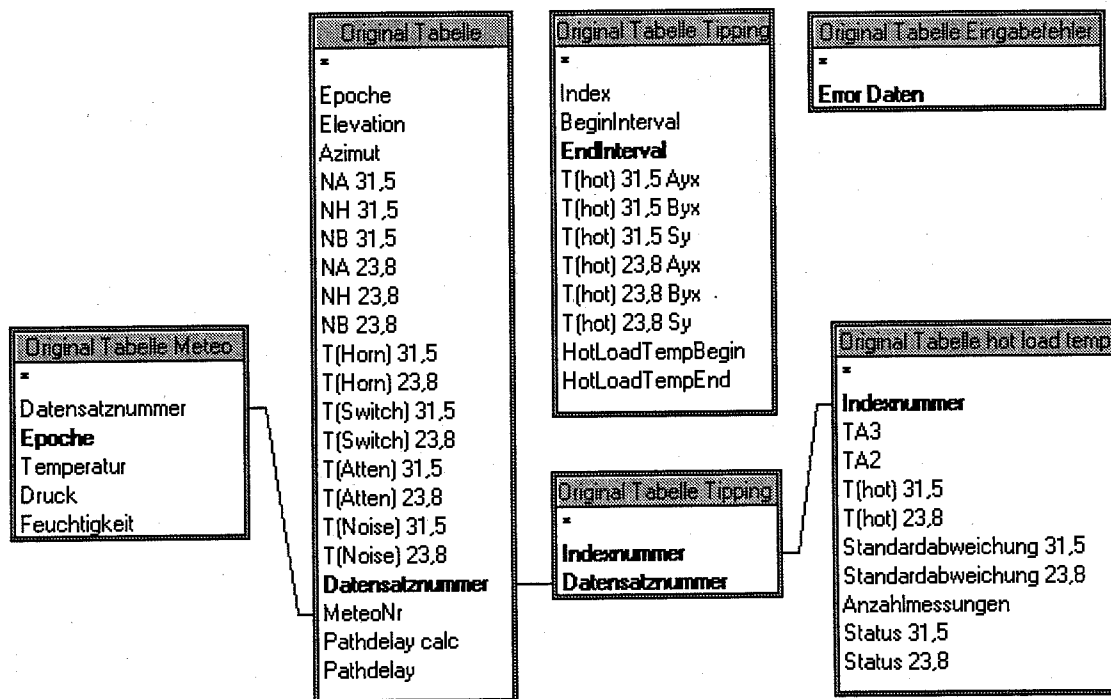
### 6.1 Aufbau der WVR Datenbank

Für das GGL-IGP Radiometer wird die MS-ACCESS-Datenbank unter dem Betriebssystem MS-DOS/ Windows verwendet (Hoffbauer, 1992). Die gemessenen WVR-Daten werden in eine ASCII-Datei gespeichert und anschliessend in die Datenbank eingelesen. Innerhalb der Datenbank werden alle Datensätze in Tabellen gespeichert und verwaltet. Für das WVR werden die folgenden Tabellen pro Messreihe benötigt.

- Grundtabelle (alle gemessenen Daten)
- Eingabefehler (alle unvollständigen und fehlerhafte Datensätze)
- Heisslasttemperaturen (eine Heisslasttemperaturen pro Tipping Curve)
- Meteodaten
- Lineare Regression der Tipping Curve pro Zeitintervall
- Tipping to Data (Verbindungstabelle Daten zu Heisslasttemperaturen)

Pro Datengruppe werden jeweils diese sechs Tabellen mit den entsprechenden Verknüpfungen angelegt.

#### 6.1.1 Verknüpfungen der einzelnen Tabellen



Figur 15 : Verknüpfungen der einzelnen Tabellen innerhalb einer Datengruppe

In der **Grundtabelle** sind alle vom WVR erfassten Daten, Verknüpfungsfelder zu anderen Tabellen sowie der berechnete Path Delay abgelegt.

### **Eingabefehler:**

Treten beim Einlesen von WVR-Daten Fehler auf, so wird der jeweilige fehlerhafte Datensatz in ASCII-Format in dieser Tabelle gespeichert.

### **Heisslasttemperatur (Hot Load Temperature) :**

Für die Kalibration des WVR müssen die Heisslasttemperaturen berechnet werden. Damit für die Glättung der Messreihe die Daten nicht erneut berechnet werden müssen, werden sie in dieser Tabelle gespeichert und mit den originalen Daten verknüpft.

### **Meteo :**

Diese Tabelle enthält die meteorologischen Daten Temperatur, Druck und relative Luftfeuchtigkeit der Stationsumgebung, die während der Messperiode mit dem Meteo-Datenlogger aufgenommen wurden.

### **Lineare Regression der Tipping Curve :**

Bei der Berechnung des Path Delay werden Heisslasttemperaturen benötigt (siehe Seite 20). Die Zeitintervalle der linearen Regression der Heisslasttemperaturen werden in einem ersten Schritt vom Programm automatisch bestimmt und können später in der Nachbearbeitung manuell verändert werden

### **Tipping to Data :**

Diese Tabelle enthält die Verknüpfungen der Grundtabelle mit der Heisslasttemperaturen-Tabelle. Es ist zu bemerken, dass in einer relationalen Datenbank die gleichen Daten nur einmal vorhanden sein dürfen, d.h. die Datenbank muss normalisiert sein.

Die einzelnen Felder der Datenbank sind im Anhang A beschrieben.

## 6.2 Anleitung zur WVR-Auswertesoftware

### 6.2.1 Einleitung

In dieser Anleitung wird beschrieben, wie mit der Auswertesoftware zum WVR die gemessenen WVR-Daten in die Datenbank aufgenommen, ausgewertet und graphisch dargestellt werden können. Die Auswertesoftware ist in der Datenbank ACCESS unter MS-Windows entwickelt worden. Deshalb werden minimale Kenntnisse der MS-Windows-Umgebung vorausgesetzt. Eine Einführung in MS-Windows finden Sie zum Beispiel im Microsoft® Windows 1990-1992, Benutzerhandbuch, Kapitel Grundlagen. Innerhalb von ACCESS kommt ein Hauptfenster als Regiezentrum zum Einsatz. In diesem können Sie aus Listen die gewünschten Daten und Befehle auswählen und mit Knöpfen die entsprechenden Funktionen auswählen. Alle anderen Datenbankfunktion von ACCESS (ACCESS, 1992) können ebenfalls verwendet werden (Hoffbauer, 1992). Alle Befehle zum Bedienen der WVR-Auswertesoftware können im Hauptfenster gewählt und ausgeführt werden. Zum Auswählen von Daten und Funktionen können Tastenkürzel<sup>5</sup> oder die Maus verwendet werden.

- Kapitel 1 beschreibt die Startprozedur für die WVR-Datenbank.
- Kapitel 2 enthält Informationen zum Hauptmenü / Hauptfenster der WVR Auswertesoftware.
- Kapitel 3 erklärt den Vorgang zum Einlesen von Daten in die WVR-Datenbank.
- Kapitel 4 beschreibt die Möglichkeiten der Datenauswertung.
- Kapitel 5 zeigt die verschiedenen Auswahlmöglichkeiten bei der Visualisierung der WVR-Daten.
- Kapitel 6 enthält Informationen zum Aufbau von neuen Tabellen innerhalb der WVR-Datenbank.

---

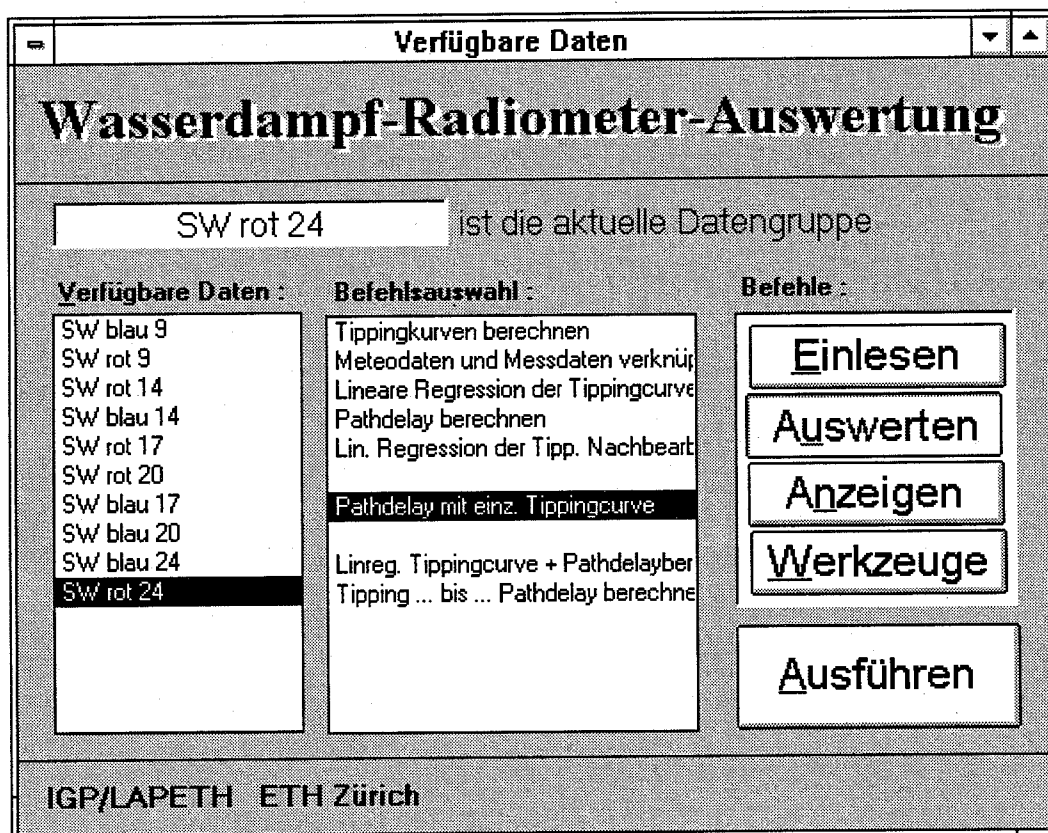
<sup>5</sup>Tastenkürzel sind Tastenkombinationen, die der Anwender drücken kann, um in einer Anwendung eine bestimmte Aufgabe auszuführen. Tastenkürzel können grundsätzlich parallel zu Menüfunktionen verwendet werden.

## 6.2.2 Starten der WVR-Datenbank

Zuerst muss MS-ACCESS, die Datenbankapplikation selbst, innerhalb von MS-Windows aufgestartet werden. Dies geschieht in der Umgebung des Programmanagers von MS-Windows (Microsoft Windows Benutzerhandbuch, 1990-1992). Ist ACCESS aufgestartet, kann unter dem Menüpunkt Datei, Datenbank öffnen... die entsprechende WVR-Datenbank innerhalb von ACCESS geöffnet werden (Hoffbauer, 1992). Beim Aufstarten einer WVR-Datenbank wird das Hauptfenster mit den verfügbaren Daten automatisch eröffnet (siehe Seite 23).

## 6.2.3 Hauptfenster / Hauptmenü

Um die WVR-Daten möglichst einfach und bedienerfreundlich zu verarbeiten, wurde für die Datenbank eine menügesteuerte Oberfläche programmiert. Die Figur 16 zeigt das Hauptfenster der WVR-Auswertesoftware. In diesem Formular können die nötigen Manipulationen in der WVR-Datenbank vorgenommen werden.



Figur 16 : Hauptfenster der WVR-Auswertesoftware

- Verfügbare Daten:** In dieser Liste sind alle in der Datenbank verfügbaren WVR-Datengruppen (Datenfiles) aufgelistet. Die jeweils aktuelle Datengruppe erscheint oberhalb der Liste in einem Feld. Wichtig: Es muss immer eine Datengruppe ausgewählt sein.
- Befehlsauswahl:** Je nach selektiertem Befehl erscheint in dieser Liste eine Auswahl an Unterbefehlen zum jeweiligen Befehl.
- Befehle:** Hier kann einer der vier Hauptbefehle selektiert werden: Einlesen, Auswerten, Anzeige oder Werkzeug auswählen.

**Ausführen :** Mit diesem Knopf wird der selektierte Befehl mit dem ebenfalls ausgewählten Unterbefehl ausgeführt.

Wird der Ausführungsknopf in Figur 16 mit der Maus oder den Tastenkürzel <Alt><A> gedrückt, so wird in der Datengruppe "Schweden rot 24" der Path Delay mit den jeweiligen Tipping Curve Werten (ohne lineare Regression) berechnet. In Anhang B auf Seite 40 ist die Funktionsweise des Hauptformulars erklärt.

Diagramm 1 ist eine Übersicht über alle in der WVR-Datenbank direkt über das Hauptformular anwählbaren Funktionen. Alle anderen Funktionen, wie spezielle Abfragen oder Datenexport, müssen in ACCESS einzeln ausgeführt werden.

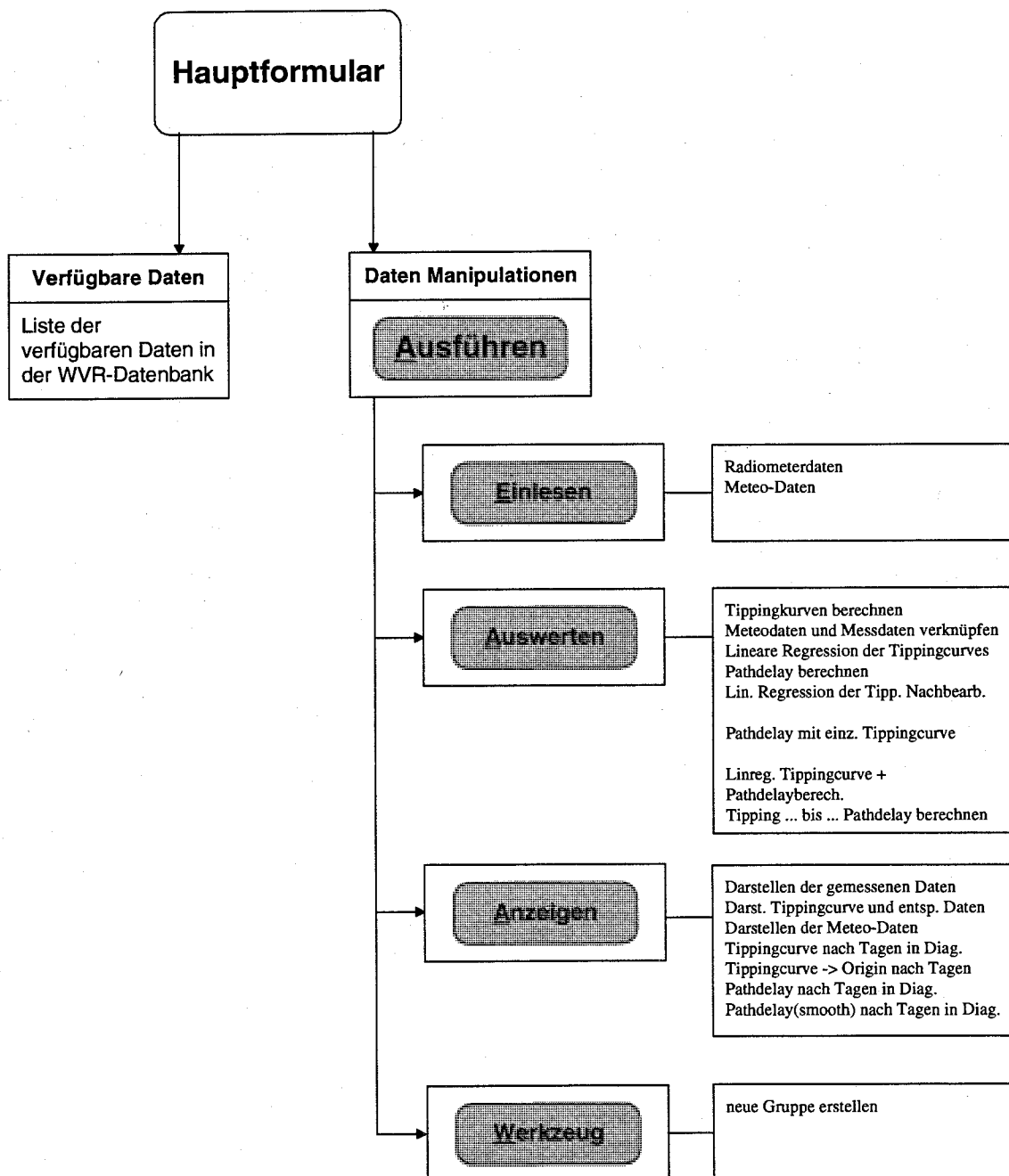


Diagramm 1 : Übersicht WVR-Datenbank Manipulationen

## 6.2.4 Einlesen

Die Daten des WVR werden mit einem PC aufgezeichnet und in einem ASCII-File gespeichert. Mit dem Befehl **Einlesen / Radiometerdaten** können diese Daten in die WVR-Datenbank übertragen werden. Treten während der Aufnahme der Daten Fehler auf, d.h. ist ein Datensatz fehlerhaft, so wird dieser in die Tabelle "... Eingabefehler" geschrieben. Er kann allenfalls manuell korrigiert und dann erneut eingelesen werden. Für eine genaue Programmbeschreibung siehe Anhang C Seite 43.

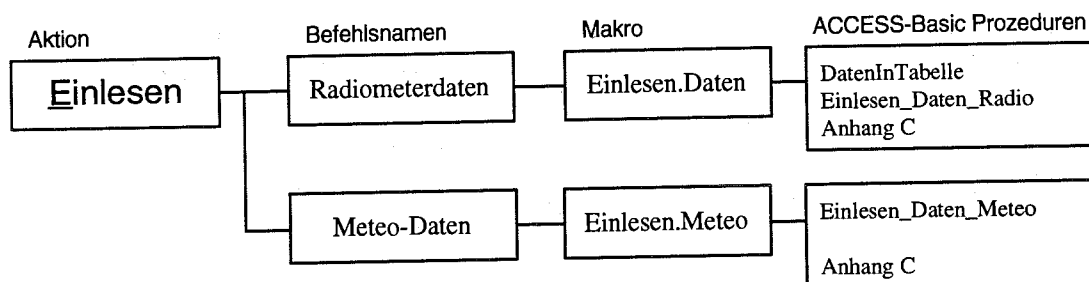


Diagramm 2 : Übersicht WVR-Datenbank-Befehle zum Einlesen der Daten

Die meteorologischen Daten können mit dem Befehl **Einlesen / Meteo-Daten** in die Datenbank übertragen werden. Sind sowohl WVR-Daten als auch die Meteodaten eingelesen, kann mit der Verarbeitung der Daten begonnen werden. Für die Verarbeitung ist es wichtig, dass zu den WVR-Daten die Meteowerte verfügbar sind. Diese können entweder von den am Messsystem direkt angeschlossenen Meteorloggern oder von anderen Quellen übernommen werden, wie z. B. von VLBI-Stationen, Flugplätzen oder anderen meteorologischen- oder geodätischen Observatorien.



### 6.2.5 Auswerten

Sind die WVR-Daten in der Datenbank aufgenommen, kann mit der Auswertung begonnen werden. Um die Anzeige der Daten zu beschleunigen, werden die Zwischenergebnisse jeweils wieder in Tabellen gespeichert. Um den Path Delay berechnen zu können, muss die folgende Reihenfolge der Verarbeitung eingehalten werden:

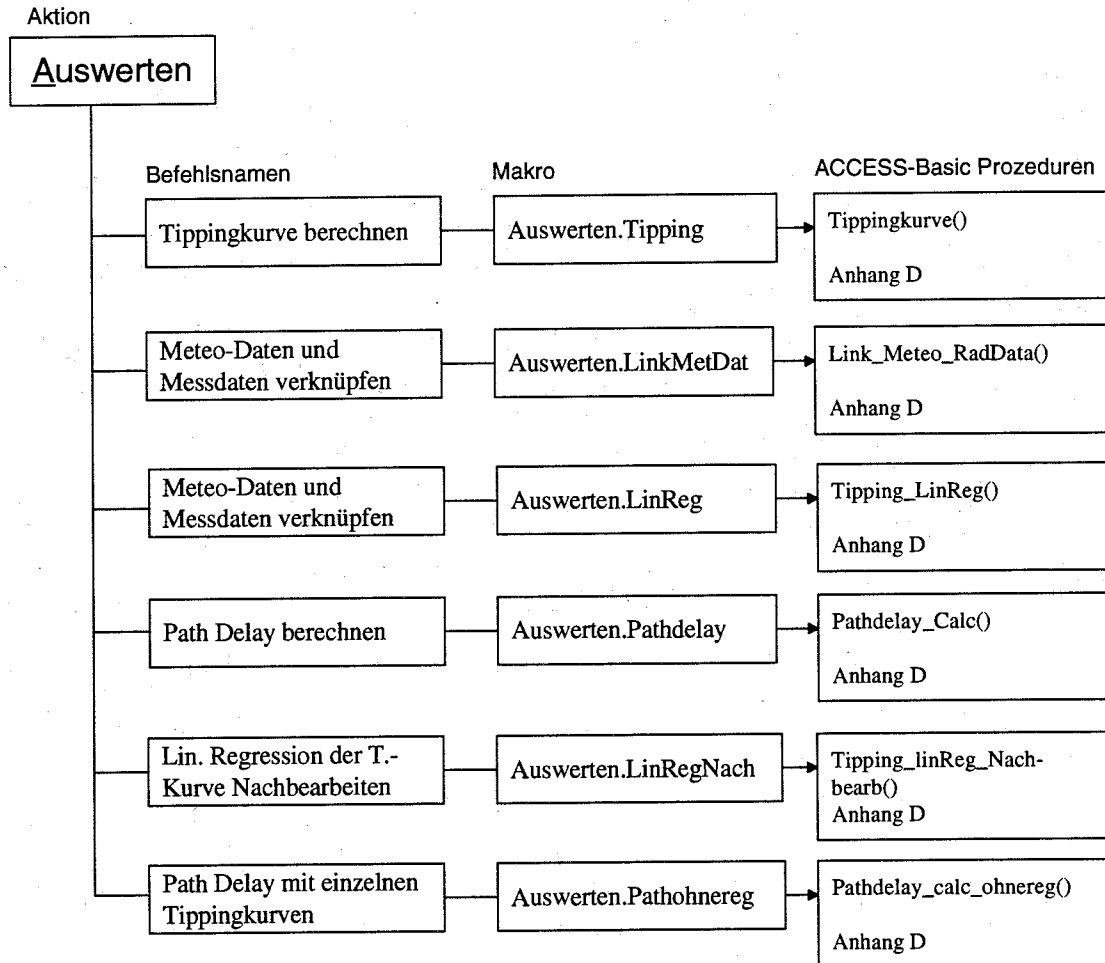


Diagramm 3 : Übersicht WVR-Datenbank-Befehle Auswerten der Daten

Mit dem Befehl "Tippingkurvenberechnen" werden die einzelnen Tipping Curves nach den Formeln (7) bis (11) Seite 21 berechnet. Die Resultate werden in der Tabelle "... Tipping" gespeichert und mit der Originaltabelle verknüpft, damit jederzeit zu einer Tipping Curve die entsprechenden Messdaten gefunden werden können.

Die meteorologischen Daten müssen nach dem Einlesen noch mit den Messdaten verknüpft werden. Zu jedem Messwert wird der zeitlich am nächsten liegende Meteowert gesucht und die entsprechende Datensatznummer des Meteowertes in die Grundtabelle eingetragen. Nur wenn der Meteowert vorhanden ist, kann nach der Formel (3) der Path Delay berechnet werden.

Nach der Berechnung der Heisslastkorrekturen muss über die ganze Heisslasttemperaturmessreihe eine lineare Regression durchgeführt werden (s.

Heisslastkorrektur Seite 21). Sind die Meteodaten vorhanden und die lineare Regression der Heisslasttemperatur durchgeführt, kann die linearisierte Helligkeitstemperatur (Formel (3) mit  $T_B$  nach Formel(5)) und damit der Path Delay (Formel(4)) berechnet werden (siehe Seite 4).

Eine Heisslasttemperaturmessreihe kann durch verschiedene Einflüsse, wie z.B. Regen, Unstetigkeitsstellen aufweisen. Damit die lineare Regression möglichst genau ist, muss die Messreihe unterbrochen werden. In der Ansicht "Tipping Curve nach Tagen in Diagramm" kann das Messreihenintervall neu getrennt werden (siehe Seite 33).

Mit dem Befehl "Path Delay mit einzelnen Tipping Curve" kann der Path Delay, ohne die lineare Regression der Heisslastkorrektur zu berücksichtigen, berechnet werden. In Formel (5) wird für die Heisslasttemperatur  $T_H$  des entsprechenden Datensatzes verwendet.

Mit den anderen beiden Befehlen werden jeweils mehrere Aktionen nacheinander ausgeführt. So kann mit "Tipping ... bis ... Path Delay" eine komplette Verarbeitung der WVR-Daten durchgeführt werden.

## 6.2.6 Anzeigen

Mit den Befehlen der Gruppe Anzeigen können die Daten in ACCESS als numerische Werte oder direkt graphisch dargestellt werden. Mit dem Programm MS-Graph wird die Graphik in ACCESS in verschiedenen Formularen angezeigt.

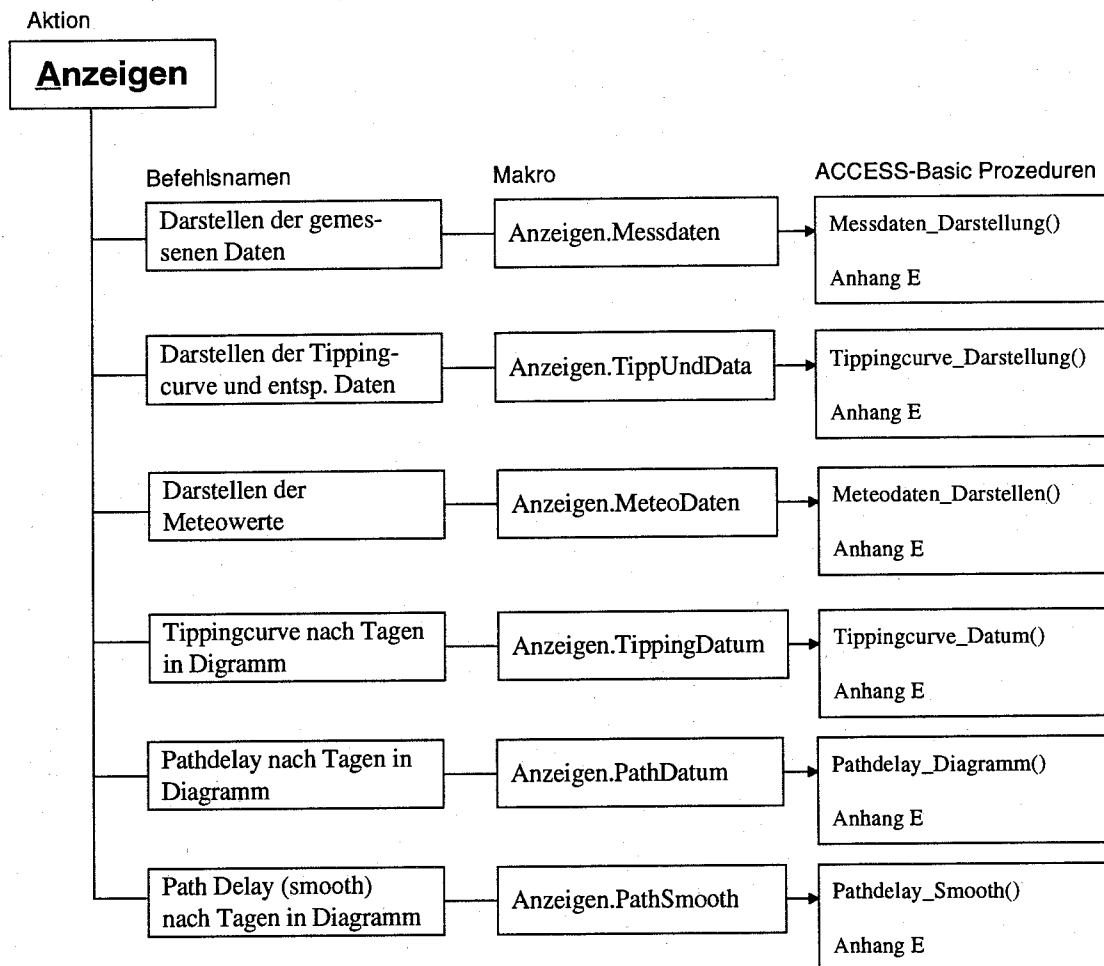


Diagramm 4 : Übersicht WVR-Datenbank-Befehle Anzeigen der Daten

Die gemessenen WVR-Daten können mit dem Formular (Figur 17) dargestellt und verändert werden. In diesem Formular werden keine Verknüpfungen zu anderen Daten angezeigt. Um Verknüpfungen darzustellen, muss in ACCESS unter Tabellen die Grundtabelle der WVR-Datenbank aufgerufen werden.

Darstellung der Messdaten								
Darstellung der Messdaten								
Epoche	Elevation	Azimut	NA 31,5	NH 31,5	NB 31,5	NA 23,8	NH 23,8	
23.10.92 14:33:54	30	0	6.4532	7.6121	5.1941	4.7726	7.6988	
23.10.92 14:34:03	30	10	6.4549	7.6136	5.1919	4.775	7.7008	
23.10.92 14:34:13	30	20	6.4522	7.6131	5.1882	4.775	7.7115	
23.10.92 14:34:22	30	30	6.4524	7.6139	5.1795	4.7752	7.7093	
23.10.92 14:34:32	30	40	6.4546	7.6109	5.1897	4.7762	7.7105	
23.10.92 14:34:42	30	50	6.4485	7.6134	5.1917	4.7762	7.7125	
23.10.92 14:34:51	30	60	6.4497	7.6131	5.1851	4.7818	7.7125	
23.10.92 14:35:01	30	70	6.451	7.6136	5.2014	4.7765	7.7113	
23.10.92 14:35:10	30	80	6.4497	7.6126	5.1778	4.7804	7.714	
23.10.92 14:35:20	30	90	6.4537	7.6097	5.1941	4.7818	7.7215	
23.10.92 14:35:30	30	100	6.4532	7.6134	5.177	4.784	7.7225	
23.10.92 14:35:39	30	110	6.45	7.608	5.1663	4.7845	7.7247	
23.10.92 14:35:49	30	120	6.45	7.6097	5.1795	4.7867	7.7289	
23.10.92 14:35:58	30	130	6.452	7.6112	5.1697	4.7889	7.7311	
23.10.92 14:36:08	30	140	6.45	7.6141	5.1897	4.7887	7.7308	

Figur 17: Gemessene Originaldaten des Radiometers

Oft ist es nützlich, zu einer Heisslasttemperatur resp. Heisslastkorrektur die entsprechenden Daten der linearen Regression darzustellen. Im Formular wird jeweils eine Tipping Curve mit ihren Daten angezeigt.

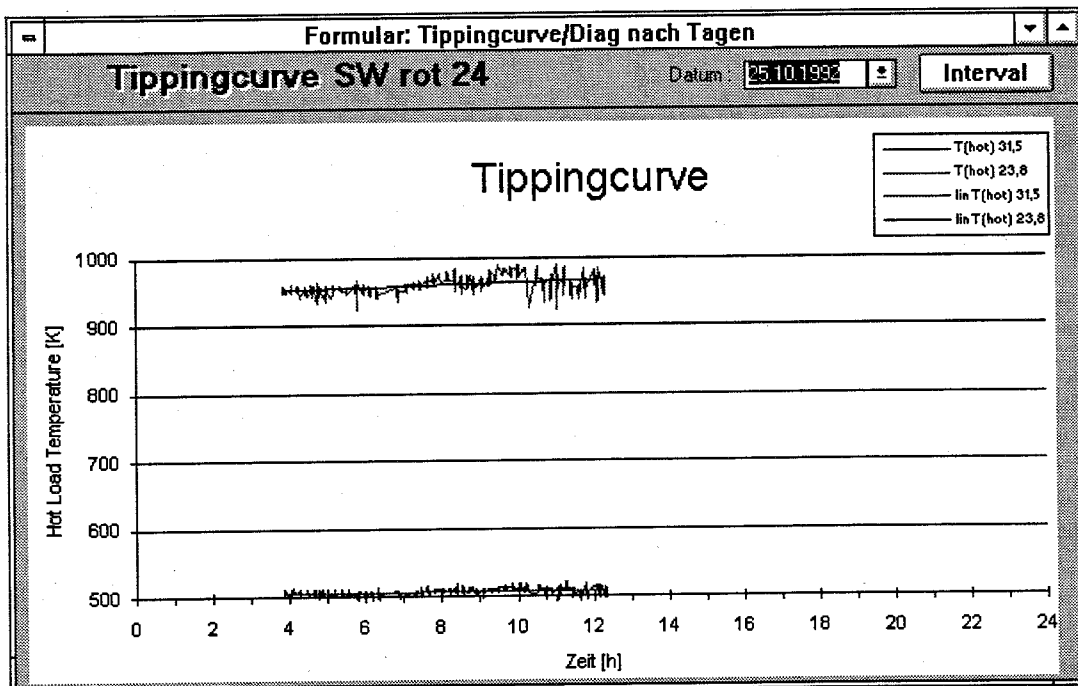
Hot load temp								
Hot Load Temperature								
Indexnummer:	913	I(hot) 31,5:	496,0	TA 31,5:	282,5			
Anzahlmessungen:	3	I(hot) 23,8:	934,3	TA 23,5:	282,6			
Azimut:	315,0	Standardabweichung 31,5:	11,93069					
Messnummer und Daten		Standardabweichung 23,8:	14,12003					
Epoche	Elevation	Azimut	NA 31,5	NH 31,5	NB 31,5	NA 23,8	NH 23,8	
23.10.92 14:40:55	23,6	315	6.4503	7.6124	5.2546	4.7975	7.7357	
23.10.92 14:41:04	30	315	6.4515	7.6112	5.1753	4.7962	7.7364	
23.10.92 14:41:14	41,8	315	6.4463	7.6041	5.1275	4.7940	7.7364	

Figur 18: Heisslastkorrektur mit den entsprechenden Daten

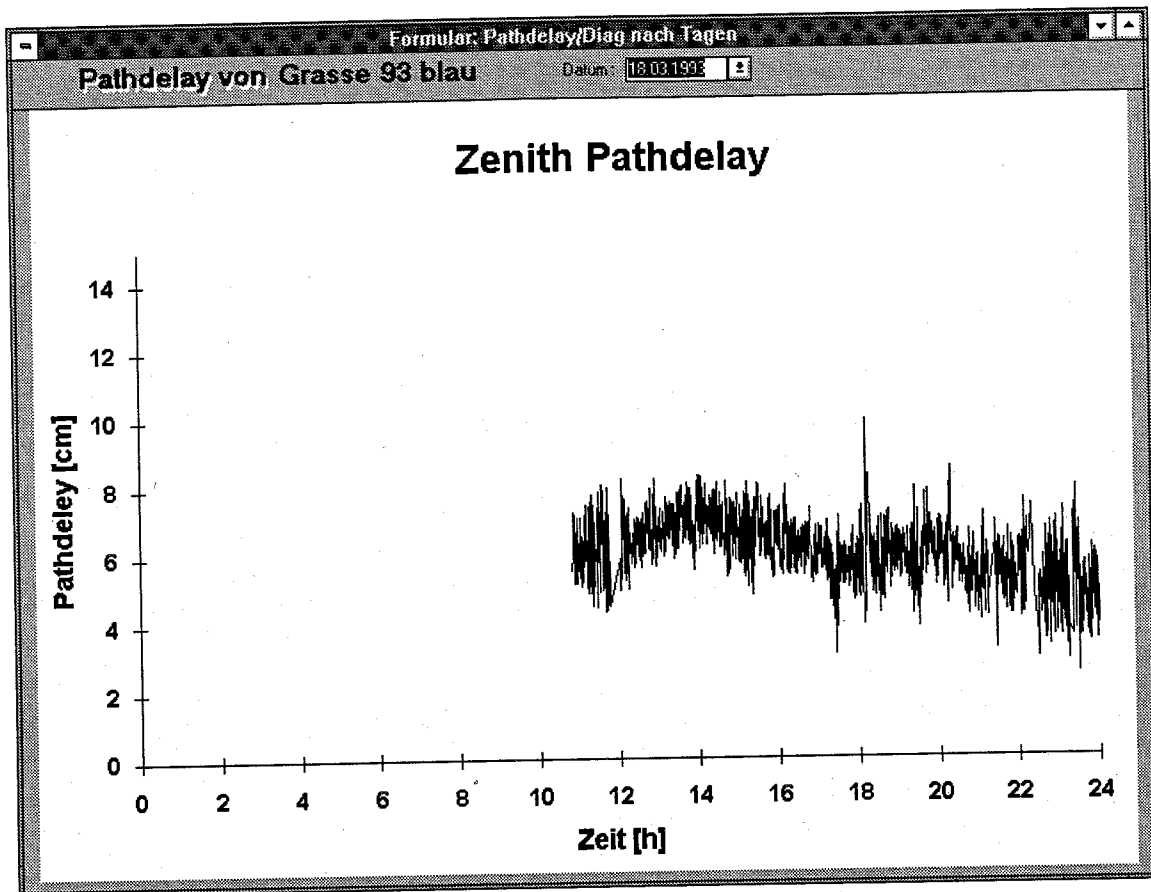
Tabelle der gemessenen Meteo-Daten				
Epoche	Temperatur [°C]	Druck [hPa]	Feuchtigkeit [%]	
20.10.92 23,16,00	5.1	1'004.0	97.4	
20.10.92 23,20,00	4.9	1'004.0	96.5	
20.10.92 23,23,00	5.1	1'004.0	97.3	
20.10.92 23,26,00	4.7	1'004.0	98.4	
20.10.92 23,29,00	4.7	1'004.0	98.1	
20.10.92 23,32,00	4.4	1'004.0	97.3	
20.10.92 23,35,00	4.4	1'004.0	97.0	
20.10.92 23,38,00	4.3	1'004.0	97.5	
20.10.92 23,41,00	4.2	1'004.0	97.6	
20.10.92 23,44,00	4.1	1'004.0	97.6	
20.10.92 23,47,00	4.0	1'004.0	99.2	
20.10.92 23,51,00	3.9	1'004.0	99.4	
20.10.92 23,54,00	3.7	1'004.0	99.6	

Figur 19: Gemessene Meteodaten

Die Heisslasttemperatur bzw. Tipping Curve kann in einem Formular graphisch dargestellt werden. Zusätzlich können mit dem Knopf "Intervall" die Zeitintervalle der Messreihe manuell neu unterteilt werden.



Figur 20: Zeitlicher Verlauf der Heisslasttemperatur

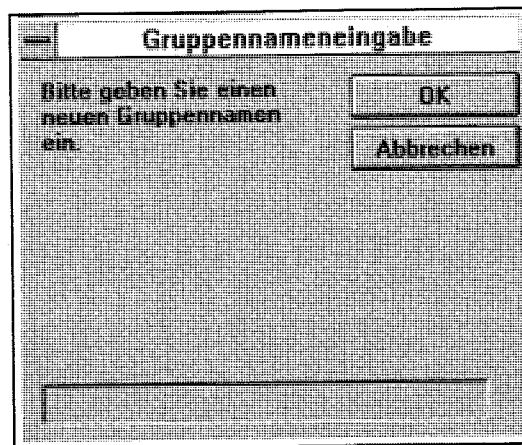


Figur 21: Path Delay der Messreihe "Schweden rot" vom 18. März 1993

Beim Path Delay werden alle Messwerte zur Berechnung berücksichtigt. Zur Anzeige kommt jeweils der Path Delay Wert im Zenit. Da je nach Messreihe sehr viele Datenpunkte pro Graphik vorhanden sein können, müssen sie reduziert werden. Mit dem Befehl " Path Delay (smooth) ... " kann eine Darstellung mit reduzierter Anzahl Punkten gewählt werden.

### 6.2.7 Werkzeug

Die WVR-Daten werden in der Datenbank in Tabellen gespeichert. Damit verschiedene Messreihen in unterschiedlichen Tabellen gespeichert werden können, müssen die Tabellen zuerst neu aufgebaut werden. Mit dem Befehl "**Werkzeug / neue Gruppe erstellen**" kann eine neue Datengruppe wie in Figur 22 dargestellt aufgebaut werden.



Figur 22: Eingabedialog zum Erzeugen einer neuen Datengruppe

## 7. Literatur

- Elgered G. , Davis J.L. , Herring T.A. and Shapiro I.I. (1991) : **Geodesy by Radio Interferometry : Water Vaper Radiometry for Estimation of the Wet Delay.** Journal of Geophysical Research, Vol 96 NO B4, Pages 6541-6555, April, 10. 1991
- Elgered G. (1993) : **Tropospheric Radio-Path Delay from Ground based Microwave Radiometry.** Reprint from "Atmospheric Remote Sensing by Microwave Interferometry", Chapter 5, Edited by Michael A. Janssen. ISBN0-471-62891-3. J. Wiley & sons, Inc.
- Hoffbauer M. und Spielmann Ch. (1992) : **Das ACCESS Buch.** SYBEX-Verlag GmbH, Düsseldorf 1992.
- Johansson J. , Elgered G. and Davis J. (1993) : **Wet path delay algorithms for use with microwave radiometer data.** Smith and Turcotte (Eds.), Contributions of Space Geodesy to Geodynamics: Crustal Dynamics. American Geophysical Union (AGU). Geodynamics Series, Vol. 25: 81-98, Washington, D.C.
- Kuehn C. , Elgered G. , Johansson J. , Clark T. , and Roennaeng B. (1993) : **A microwave radiometer comparison and its implication for the accuracy of wet delays.** Smith and Turcotte (Eds.), Contributions of Space Geodesy to Geodynamics: Crustal Dynamics. American Geophysical Union (AGU). Geodynamics Series, Vol. 25: 99-114, Washington, D.C.
- Microsoft® Windows (1990-1992) : **Microsoft® Windows™ Benutzerhandbuch** für das Windows-Betriebssystem. © 1990-1992 Microsoft Corporation.
- Microsoft® ACCESS (1992) : **Microsoft® ACCESS Benutzerhandbuch** , Relationale Datenbank für Windows. © 1992 Microsoft Corporation.
- Microsoft® ACCESS (1992) : **Microsoft® ACCESS Einführung in die Programmierung** , Relationale Datenbank für Windows. © 1992 Microsoft Corporation.
- Waters J.W. (1976) : **Absorption and Emission by Atmospheric Gases.** Methods of Experimental Physics, Vol 12. New York Academic Press 1976.
- Wild U. (1994) : **Ionosphere and Geodetic satellite systems : Permanent GPS tracking and monitoring.** Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 48, im Druck.
- Wu S.C. (1978) : **Frequency Selection and Calibration of Water Vapour Radiometer.** DSN Progress Report 42-43, Jet Propulsion Laboratory, Pasadena, CA, pp 76-81, Feb. 15, 1978
- Wu S.C., Claflin E.S. und Resch G.M. (1978) : **Microwave Radiometer Measurement of Water Vapour Path Delay: Data Reduction techniques.** DSN Progress Report 42-48, Jet Propulsion Laboratory, Pasadena, CA, pp 22-30, September und Oktober 1978.



Wu S.C. (1979) : **Optimum Frequencies of a Passiv Microwave Radiometer for Tropospheric Path-Length Correction** , IEEE Transactions on Antennas and Propagation, Vol. AP-27, NO 2, March 1979.

## Anhang A

Original Tabelle			
Feldname	Datentyp	Länge	Indexname
Epoche	Datum/Zeit	8	Epoche
Elevation	Single	4	
Azimut	Single	4	
NA 31,5	Single	4	
NH 31,5	Single	4	
NB 31,5	Single	4	
NA 23,8	Single	4	
NH 23,8	Single	4	
NB 23,8	Single	4	
T(Hom) 31,5	Single	4	
T(Hom) 23,8	Single	4	
T(Switch) 31,5	Single	4	
T(Switch) 23,8	Single	4	
T(Atten) 31,5	Single	4	
T(Atten) 23,8	Single	4	
T(Noise) 31,5	Single	4	
T(Noise) 23,8	Single	4	
Datensatznummer	Long Integer	4	PrimaryKey
MeteoNr	Long Integer	4	
Pathdelay calc	Ja/Nein	1	
Pathdelay	Single	4	

Original Tabelle Eingabefehler			
Feldname	Datentyp	Länge	Indexname
Error Daten	Text	200	PrimaryKey

Original Tabelle hot load temp			
Feldname	Datentyp	Länge	Indexname
Indexnummer	Long Integer	4	PrimaryKey
TA3	Single	4	
TA2	Single	4	
T(hot) 31,5	Single	4	
T(hot) 23,8	Single	4	
Standardabweichung 31,5	Single	4	
Standardabweichung 23,8	Single	4	
Anzahlmessungen	Byte	1	
Status 31,5	Ja/Nein	1	
Status 23,8	Ja/Nein	1	

Original Tabelle Meteo			
Feldname	Datentyp	Länge	Indexname
Datensatznummer	Long Integer	4	
Epoche	Datum/Zeit	8	PrimaryKey
Temperatur	Single	4	
Druck	Single	4	
Feuchtigkeit	Single	4	

Tabelle 1 : Felderdefinitionen der WVR-Datenbank Tabellen

<b>Original Tabelle Tipping LinReg</b>			
Feldname	Datentyp	Länge	Indexname
Index	Long Integer	4	
BeginInterval	Datum/Zeit	8	
EndInterval	Datum/Zeit	8	PrimaryKey
T(hot) 31,5 Ayx	Double	8	
T(hot) 31,5 Byx	Double	8	
T(hot) 31,5 Sy	Double	8	
T(hot) 23,8 Ayx	Double	8	
T(hot) 23,8 Byx	Double	8	
T(hot) 23,8 Sy	Double	8	
HotLoadTempBegin	Long Integer	4	
HotLoadTempEnd	Long Integer	4	
<b>Original Tabelle Tipping to Data</b>			
Feldname	Datentyp	Länge	Indexname
Indexnummer	Long Integer	4	PrimaryKey
Datensatznummer	Long Integer	4	PrimaryKey

*Tabelle 2 : Felderdefinitionen der WVR-Datenbank Tabellen*

## Anhang B

Das Hauptfenster der WVR-Auswertesoftware ist das Regiezentrum der ganzen Applikation. Es wurde mit Hilfe des Formulareditors in ACCESS erstellt (Hoffbauer, 1992). Damit das Formular bei jedem Start automatisch geöffnet wird, ist ein Makro mit dem Namen "AUTOEXEC" nötig. Als Aktion öffnet dieser Makro das Formular "Hauptfenster".

Formularkopf		
<b>Wasserdampf-Radiometer-Auswertung</b>		
Detailbereich		
=[AktData].Spalte(1) ist die aktuelle Datengruppe		
<b>Verfügbare Daten :</b>	<b>Befehlsauswahl :</b>	<b>Befehle :</b>
Ungebunden	Ungebunden	Einlesen
		Auswerten
		Anzeigen
		Werkzeuge
		Ausführen
Formularfuß		
IGP/LAPETH ETH Zürich		

Figur 23: Hauptformular Entwurfsansicht

In der nachfolgenden Tabelle "Menüs" sind alle Verknüpfungen zwischen dem Hauptfenster und den einzelnen Makros bzw. den Programmprozeduren enthalten (s. Verknüpfungsdigramm).

Die Spalte "Menu Category" entspricht jeweils dem Optionswert der Knöpfe Einlesen, Auswerten, Anzeigen und Werkzeug. Die "Item Number" gibt die Zeile und der "Menu Text" den anzuzeigenden Text im Feld Befehlsauswahl wieder. "Macro To Run" ist diejenige Spalte, die den Makro enthält, der bei aktiver Selektion durch ein Doppelklicken mit der Maustaste oder durch drücken der Ausführungstaste ausgelöst wird. Es ist zu Bemerkem, dass Programm-Prozeduren nicht direkt, sondern nur via Makro ausgeführt werden können.

MenuCategory	ItemNumber	MenuText	MacroToRun
1	1	Radiometerdaten	Einlesen.Daten
1	2	Meteo-Daten	Einlesen.Meteo
2	1	Tippingkurven berechnen	Auswerten.Tipping
2	2	Meteodaten und Messdaten verknüpfen	Auswerten.LinkMetDat
2	3	Lineare Regression der Tippingcurves	Auswerten.LinReg
2	4	Pathdelay berechnen	Auswerten.Pathdelay
2	5	Lin. Regression der Tipp. Nachbearb.	Auswerten.LinRegNach
2	6		
2	7	Pathdelay mit einz. Tippingcurve	Auswerten.Pathohnereg
2	8		
2	9	Linreg. Tippingcurve + Pathdelayberech.	Auswerten.LinRegPath
2	10	Tipping ... bis ... Pathdelay berechnen	Auswerten.DoAll
3	1	Darstellen der gemessenen Daten	Anzeigen.Messdaten
3	2	Darst. Tippingcurve und entsp. Daten	Anzeigen.TippUndData
3	3	Darstellen der Meteo-Daten	Anzeigen.MeteoDaten
3	4	Tippingcurve nach Tagen in Diag.	Anzeigen.TippingDatum
3	5	Tippingcurve -> Origin nach Tagen	Anzeigen.TippingOrigin
3	6	Pathdelay nach Tagen in Diag.	Anzeigen.PathDatum
3	7	Pathdelay (smooth) nach Tagen in Diag.	Anzeigen.PathSmooth
4	1	neue Gruppe erstellen	Werkzeuge.neue_Gruppe

Tabelle 3: Verknüpfungstabelle Hauptfenster zu den einzelnen Makros

Die nachfolgenden zwei Tabellen enthalten alle Steuerelementnamen des Hauptformulars und ihre wichtigsten Eigenschaften.

Steuerelementname	Beschriftung	Datensatzherkunft	Aktiviert	BeimKlicken
Text7	Wasserdampf-Radiometer-Auswertung		0	
Text8	Wasserdampf-Radiometer-Auswertung		0	
DatenName			0	
Text12	Aktuelle Datenauswahl :		0	
AktData		Verfügbare Daten	-1	
Text14	Verfügbare Daten :		0	
Text16	ist die aktuelle Datengruppe		0	
MenuItems		ListMenuOptions	-1	
MenuCat			-1	
Text21	Befehle :		0	
Btn1	Einlesen		-1	
Btn2	Auswerten		-1	
Btn3	Anzeigen		-1	
Btn4	Werkzeuge		-1	
Text26	Befehlsauswahl :		0	
BtnSelect	&Ausführen		-1	Hauptfenster.DoItem
Text27	IGP/LAPETH ETH Zürich		0	

Tabelle 4: Elementdefinitionen des Hauptformulars der WVR-Datenbank

Steuerelementname	Herkunftstyp	NachAktualisierung	Optionswert	Steuerelementinhalt
Text7				
Text8				
DatenName				=[AktData].Column(1)
Text12				
AktData	Table/Query	Hauptfenster.FeldUpdate		
Text14				
Text16				
MenuItems	Table/Query			
MenuCat		Hauptfenster.Requery		
Text21				
Btn1			1	
Btn2			2	
Btn3			3	
Btn4			4	
Text26				
BtnSelect				
Text27				

*Tabelle 5: Elementdefinitionen des Hauptformulars der WVR-Datenbank*

## Anhang C

### Makro Einlesen

Um die Prozeduren des Moduls Einlesen ausführen zu können, müssen zuerst die entsprechenden Makros aufgerufen werden.

Makroname : Einlesen			
Makrogruppe	Aktion	Bedingung	Argument1
Daten	S onduhr		Ja
	AusführenCode		Einlesen_Daten_Radio()
	S onduhr		Nein
Meteo	S onduhr		Ja
	AusführenCode		Einlesen_Daten_Meteo()
	S onduhr		Nein

*Tabelle 6: Makrogruppe Einlesen*

### Modul Einlesen

#### Deklarationen im Modul Einlesen

```
Option Compare Database      'Verwenden der Datenbank-
                             'Sortierreihenfolge beim
                             'Vergleich von Zeichenfolgen.

Option Explicit

Dim DB As Database
Dim Tabelle As Table
Dim ErrorTab As Table
Dim SBuff As String

Dim inf As Variant
```

## Prozedur Einlesen der meteorologischen Daten in die Tabelle "... Meteo"

```

Function Einlesen_Daten_Meteo ()

Dim FileName, GruppenName As String
Dim InText, OutText As String
Dim SteuerElement As Control
Dim hWnd, StatusDialog, StatusEditor As Integer

Rem Begin Function
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
Set Tabelle = DB.OpenTable(GruppenName + " Meteo")
hWnd = wu_GetActiveWindow(): 'Filename erfragen
InText = "*.met"
OutText = "
StatusDialog = FileNameFragen(hWnd, InText, OutText) 'in DLL-Modul
FileName = Mid$(OutText, 1)
Do While StatusDialog = 1
Open FileName For Input As #1
Do While (Not EOF(1) And StatusDialog = 1)
Line Input #1, SBuff
If ((Val(Mid$(SBuff, 48, 5)) > 0) And
(Val(Mid$(SBuff, 48, 5)) < 120)) Then
Tabelle.AddNew
Tabelle.[Epoche] = CVDate(Val(Mid$(SBuff, 7, 14)) - 15018)
Tabelle.[Temperatur] = Val(Mid$(SBuff, 34, 6))
Tabelle.[Druck] = Val(Mid$(SBuff, 41, 5))
Tabelle.[Feuchtigkeit] = Val(Mid$(SBuff, 48, 5))
Tabelle.Update
Else
If IsDate(Mid$(SBuff, 1, 19)) Then
Tabelle.AddNew
Tabelle.[Epoche] = CVDate(Mid$(SBuff, 1, 19))
Tabelle.[Temperatur] = Val(Mid$(SBuff, 29, 6))
Tabelle.[Druck] = Val(Mid$(SBuff, 21, 6))
Tabelle.[Feuchtigkeit] = Val(Mid$(SBuff, 40, 6))
Tabelle.Update
Else
Rem *** Datensatz hat nicht Format Datum/Zeit Temp. Druck Feucht
End If
End If
Tabelle.MoveLast
DoCmd Echo True, "Eingelesene Datensätze = "+Tabelle.Datensatznummer
Loop
Close #1
OutText = "
StatusDialog = FileNameFragen(hWnd, InText, OutText)
FileName = Mid$(OutText, 2)
Loop
DB.Close
End Function

```



## Prozedur Einlesen der WVR-Daten in die Tabelle "..." (Haupttabelle)

```
Function Einlesen_Daten_Radio ()

Dim FileName, GruppenName As String
Dim InText, OutText As String
Dim SteuerElement As Control
Dim hWnd, StatusDialog, StatusEditor As Integer

Rem Begin Function
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
Set Tabelle = DB.OpenTable(GruppenName)
Set ErrorTab = DB.OpenTable(GruppenName + " Eingabefehler")
hWnd = wu_GetActiveWindow() 'Filename erfragen
InText = "*.rdm"
OutText = "
StatusDialog = FileNameFragen(hWnd, InText, OutText)
FileName = Mid$(OutText, 1)
Do While StatusDialog = 1
Open FileName For Input As #1
DoCmd Hourglass True
Do While (Not EOF(1) And StatusDialog = 1)
Line Input #1, SBuff
If ((Val(Mid$(SBuff, 133, 7)) > 275) And
(Val(Mid$(SBuff, 133, 7)) < 315)) And
(Val(Mid$(SBuff, 22, 5)) > 1) Then
DatenInTabelle
Else
Rem *** Datensatz mit "Bad" Daten in Errorfile speichern
ErrorTab.AddNew
ErrorTab.[Error Daten] = SBuff
ErrorTab.Update
End If
Loop
Close #1
DoCmd Hourglass False
OutText = "
StatusDialog = FileNameFragen(hWnd, InText, OutText)
FileName = Mid$(OutText, 2)
Loop
DB.Close
End Function
```

## Prozedur Speichern der WVR-Daten in die Tabelle "... " (Haupttabelle)

```
Sub DatenInTabelle ()
```

```
Rem Begin
```

```
  Tabelle.AddNew
```

```
    Tabelle.Epoche = CDate(Val(Mid$(SBuff, 7, 14)) - 15018)
```

```
    Tabelle.Elevation = Val(Mid$(SBuff, 22, 5))
```

```
    Tabelle.Azimut = Val(Mid$(SBuff, 29, 5))
```

```
    Tabelle.[NA 31,5] = Val(Mid$(SBuff, 35, 6))
```

```
    Tabelle.[NH 31,5] = Val(Mid$(SBuff, 42, 6))
```

```
    Tabelle.[NB 31,5] = Val(Mid$(SBuff, 49, 6))
```

```
    Tabelle.[NA 23,8] = Val(Mid$(SBuff, 56, 6))
```

```
    Tabelle.[NH 23,8] = Val(Mid$(SBuff, 63, 6))
```

```
    Tabelle.[NB 23,8] = Val(Mid$(SBuff, 70, 6))
```

```
    Tabelle.[T(Horn) 31,5] = Val(Mid$(SBuff, 77, 7))
```

```
    Tabelle.[T(Horn) 23,8] = Val(Mid$(SBuff, 85, 7))
```

```
    Tabelle.[T(Switch) 31,5] = Val(Mid$(SBuff, 93, 7))
```

```
    Tabelle.[T(Switch) 23,8] = Val(Mid$(SBuff, 101, 7))
```

```
    Tabelle.[T(Atten) 31,5] = Val(Mid$(SBuff, 109, 7))
```

```
    Tabelle.[T(Atten) 23,8] = Val(Mid$(SBuff, 117, 7))
```

```
    Tabelle.[T(Noise) 31,5] = Val(Mid$(SBuff, 125, 7))
```

```
    Tabelle.[T(Noise) 23,8] = Val(Mid$(SBuff, 133, 7))
```

```
  Tabelle.Update
```

```
  Tabelle.MoveLast
```

```
  DoCmd Echo True, "Eingelesene Datensätze = "+Tabelle.Datensatznummer
```

```
End Sub
```

## Modul DLL

Um Dynamic-Link-Library(DLL) Prozeduren von MS-Windows benützen zu können, muss sie in einem Module (DLL-Modul) deklariert werden. Mit dem Konzept der DDL's kann ACCESS-Basic mit Pascal oder C erweitert werden. Im Modul Einlesen wird der Eingabedialog (FileNameFragen) aus der DLL Filedlgs.dll verwendet. Wichtig ist, dass diese DLL immer mit der WVR-Datenbank in Windows zur Verfügung steht.

### Deklarationen im Modul Einlesen

```
Option Compare Database 'Verwenden der Datenbank-Sortierreihenfolge
beim Vergleich von Zeichenfolgen.
Option Explicit
'
' Type Wu_RECT.
'
Type Wu_RECT
  x1 As Integer
  y1 As Integer
  x2 As Integer
  y2 As Integer
End Type
'
' Windows API Declarations.
'
Declare Function wu_CheckMenuItem Lib "User" Alias "CheckMenuItem" (ByVal hMenu%, ByVal
  idEnableItem%, ByVal uEnable%) As Integer
Declare Sub wu_DrawMenuBar Lib "User" Alias "DrawMenuBar" (ByVal hWnd%)
Declare Function wu_EnableMenuItem Lib "User" Alias "EnableMenuItem" (ByVal hMenu%,
  ByVal idEnableItem%, ByVal uEnable%) As Integer
Declare Function wu_GetActiveWindow Lib "User" Alias "GetActiveWindow" () As Integer
Declare Function wu_GetClassName Lib "User" Alias "GetClassName" (ByVal hwin%, ByVal
  stBuf$, ByVal cch%) As Integer
Declare Function wu_GetClientRect Lib "User" Alias "GetClientRect" (ByVal hwin%,
  rectangle As Wu_RECT) As Integer
Declare Function wu_GetDC Lib "User" Alias "GetDC" (ByVal hw%) As Integer
Declare Function wu_GetDesktopWindow Lib "User" Alias "GetDesktopWindow" () As Integer
Declare Function wu_GetDeviceCaps Lib "GDI" Alias "GetDeviceCaps" (ByVal hDC%, ByVal
  iCapability%) As Integer
Declare Function wu_GetFocus Lib "User" Alias "GetFocus" () As Integer
Declare Function wu_GetMenu Lib "User" Alias "GetMenu" (ByVal hWnd%) As Integer
Declare Function wu_GetMenuState Lib "User" Alias "GetMenuState" (ByVal hMenu%, ByVal
  idItem%, ByVal fuFlags%) As Integer
Declare Function wu_GetParent Lib "User" Alias "GetParent" (ByVal hwin%) As Integer
Declare Function wu_GetSubMenu Lib "User" Alias "GetSubMenu" (ByVal hMenu%, ByVal nPos%)
  As Integer
Declare Function wu_GetWindow Lib "User" Alias "GetWindow" (ByVal hw%, ByVal
  nRelationship%) As Integer
Declare Function wu_GetWindowRect Lib "User" Alias "GetWindowRect" (ByVal hwin%,
  rectangle As Wu_RECT) As Integer
Declare Function wu_IsZoomed Lib "User" Alias "IsZoomed" (ByVal hWnd%) As Integer
Declare Function wu_MoveWindow Lib "User" Alias "MoveWindow" (ByVal hwin%, ByVal x%, ByVal
  y%, ByVal dx%, ByVal dy%, ByVal fRepaint%) As Integer
Declare Function wu_SetFocus Lib "User" Alias "SetFocus" (ByVal hwin%) As Integer
Declare Function wu_ShowWindow Lib "User" Alias "ShowWindow" (ByVal hWnd%, ByVal i%) As
  Integer
Declare Function FileNameFragen Lib "C:\ACCESS\DLL\filedlgs.dll" Alias "AccDoFileOpen"
  (ByVal Window%, ByVal sInBuff$, ByVal sOutBuff$) As
  Integer
Declare Function EditorDialog Lib "C:\ACCESS\DLL\filedlgs.dll" Alias "AccDoEdit" (ByVal
  Window%, ByVal sBuff$) As Integer
```

```
' Constants used in the functions above.
'
Const Wu_SW_RESTORE = 9
Const Wu_GW_HWNDFIRST = 0
Const Wu_GW_HWNDLAST = 1
Const Wu_GW_HWNDNEXT = 2
Const Wu_GW_HWNDPREV = 3
Const Wu_GW_OWNER = 4
Const Wu_GW_CHILD = 5
Const Wu_LOGPIXELSX = 88
Const Wu_LOGPIXELSY = 90

Const Wu_MF_BYCOMMAND = &H0
Const Wu_MF_BYPOSITION = &H400
Const Wu_MF_ENABLED = &H0
Const Wu_MF_GRAYED = &H1
Const Wu_MF_DISABLED = &H2
Const Wu_MF_MENUBREAK = &H40
Const Wu_MF_CHECKED = &H8
Const Wu_MF_UNCHECKED = &H0
'
' Microsoft Access Window Classes.
'
Global Const Wu_WC_ACCESS = "OMain"
Global Const Wu_WC_ACCESSDBC = "ODb"
Global Const Wu_WC_ACCESSTBL = "OTable"
Global Const Wu_WC_ACCESSQRY = "OQRY"
Global Const Wu_WC_ACCESSFRM = "OForm"
Global Const Wu_WC_ACCESSRPT = "OReport"
Global Const Wu_WC_ACCESSMAC = "OScript"
Global Const Wu_WC_ACCESSMOD = "OModule"
Global Const Wu_WC_ACCESSFRMPOPUP = "OFormPopup"
Global Const Wu_WC_ACCESSTOOLBAR = "OToolbar"
Global Const Wu_WC_ACCESSMDICLIENT = "MDIClient"

Dim Inf As Variant
```

## Anhang D

### Makro Auswerten

Um die Prozeduren des Moduls Einlesen ausführen zu können, müssen zuerst die entsprechenden Makros aufgerufen werden.

Makroname : Auswerten			
Makrogruppe	Aktion	Bedingung	Argument1
Tipping	S anduhr		Ja
	AusführenCode		Tippingcurve()
	S anduhr		Nein
LinkMetDat	S anduhr		Ja
	AusführenCode		Link_Meteo_RadData()
	S anduhr		Nein
LinReg	S anduhr		Ja
	AusführenCode		Tipping_LinReg()
	S anduhr		Nein
Pathdelay	S anduhr		Ja
	AusführenCode		Pathdelay_Cdc()
	S anduhr		Nein
LinRegPath	S anduhr		Ja
	AusführenCode		Tipping_LinReg()
	AusführenCode		Pathdelay_Cdc()
	S anduhr		Nein
DoAll	S anduhr		Ja
	AusführenCode		Tippingcurve()
	AusführenCode		Link_Meteo_RadData()
	AusführenCode		Tipping_LinReg()
	AusführenCode		Pathdelay_Cdc()
LinRegNach	S anduhr		Ja
	AusführenCode		Tipping_LinReg_Nachbearb()
	S anduhr		Nein
Pathohnereg	S anduhr		Ja
	AusführenCode		Pathdelay_cdc_ohnereg()
	S anduhr		Nein

Tabelle 7 : Makrogruppe Auswerten

## Deklarationen im Modul Auswerten

```

Option Compare Database           'Verwenden der Datenbank-
                                  'Sortierreihenfolge beim
                                  'Vergleich von Zeichenfolgen.

Option Explicit

Const MessLücke = .0418          'Lücke zwischen den Messkampagnen
Const Std_Max_31 = 15             'Standardabweichungsmaximum für 31,5 GHz
Const Std_Max_23 = 20            'Standardabweichungsmaximum für 23,8 GHz
Const Min_23 = 700                'Minimum Hot Load Temperatur for 23,8 GHz
Const Max_23 = 1100              'Maximum Hot Load Temperatur for 23,8 GHz
Const Min_31 = 400               'Minimum Hot Load Temperatur for 31,5 GHz
Const Max_31 = 700               'Maximum Hot Load Temperatur for 31,5 GHz

Type ElevDataTyp
  Ga3 As Single
  Ga2 As Single
  m As Single
  TB3 As Single
  TB2 As Single
  TA3 As Single
  TA2 As Single
End Type

Dim DB As Database
Dim DataTab As Table
Dim ResultTab As Table
Dim ConnectTab As Table
Dim LinregTab As Table
Dim Tagesmax As Dynaset

Dim SA(1 To 15) As ElevDataTyp
Dim TA3, TA2, TH1, TH3, TH2 As Double:  Rem Temperaturen
Dim St3, St2 As Single:                 Rem Stgg. der Regr.Ger.
Dim Std3, Std2 As Single:               Rem Standardabweichung
Dim TB03, TB02 As Double:               Rem Achs.abschnitt der
                                          Rem Regressionsgeraden

Dim Azimut, Elevation As Single
Dim k As Integer:                       Rem Anzahl Messungen pro Tipcurve
Dim quellsatznummer As Long

Dim inf As Variant

Const PI = 3.14159265359
Const TC = 2.8
Const T0C = 273.15
Const Ke = .95
Const A0 = -.696
Const A1 = .53
Const A2 = -.302

```

## Procedur Tipping : Berechnen der Tipping Kurve

```

Function Tippingcurve ()
Dim GruppenName As String
Dim SteuerElement As Control
Dim Abfrage As QueryDef

Rem Begin Function TippingCurve
Rem *** Variablen initialisieren ***
k = 1
THI = 1000: Rem T(hot) Anfangswert
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
Set DataTab = DB.OpenTable(GruppenName)
Set ResultTab = DB.OpenTable(GruppenName + " hot load temp")
Set ConnectTab = DB.OpenTable(GruppenName + " Tipping to Data")
Set Abfrage = DB.OpenQueryDef("Tipping Delete")
DoCmd Echo True, " Löscht alte Daten der Hot Load Temp."
Abfrage.SQL = "Delete FROM [" + GruppenName + " hot load temp];"
Abfrage.Execute
Abfrage.SQL = "Delete FROM [" + GruppenName + " Tipping to Data];"
Abfrage.Execute
Rem *** 1. Datensatz lesen und auswerten ***
DataTab.MoveFirst
Azimut = DataTab.Azimut
Elevation = DataTab.Elevation
quellsatznummer = DataTab.Datensatznummer
SA(1).Ga3=(DataTab.[NB 31,5]-DataTab.[NA 31,5])/
(DataTab.[NH 31,5]-DataTab.[NA 31,5])
SA(1).Ga2=(DataTab.[NB 23,8]-DataTab.[NA 23,8])/
(DataTab.[NH 23,8]-DataTab.[NA 23,8])
SA(1).m = 1 / Sin(DataTab.Elevation * PI / 180)
SA(1).TA3 = (DataTab.[T(Horn) 31,5] + DataTab.[T(Switch) 31,5] +
DataTab.[T(Atten) 31,5] + DataTab.[T(Noise) 31,5]) / 4
SA(1).TA2 = (DataTab.[T(Horn) 23,8] + DataTab.[T(Switch) 23,8] +
DataTab.[T(Atten) 23,8] + DataTab.[T(Noise) 23,8]) / 4
DataTab.MoveNext
Do While DataTab.EOF = False
If DataTab.Azimut = Azimut Then
If DataTab.Elevation <> Elevation Then
k = k + 1
Elevation = DataTab.Elevation
SA(k).Ga3=(DataTab.[NB 31,5]-DataTab.[NA 31,5])/DataTab.[NH 31,5]-
DataTab.[NA 31,5])
SA(k).Ga2=(DataTab.[NB 23,8]-DataTab.[NA 23,8])/DataTab.[NH 23,8]-
DataTab.[NA 23,8])
SA(k).m = 1 / Sin(DataTab.Elevation * PI / 180)
SA(k).TA3 = (DataTab.[T(Horn) 31,5]+DataTab.[T(Switch) 31,5]+
DataTab.[T(Atten) 31,5]+DataTab.[T(Noise) 31,5])/4
SA(k).TA2 = (DataTab.[T(Horn) 23,8]+DataTab.[T(Switch) 23,8]+
DataTab.[T(Atten) 23,8]+DataTab.[T(Noise) 23,8])/4
Else

```

```
    If k > 2 Then
      Rechnen
      Speichern
    End If
    Initprocedure
  End If
Else
  If k > 2 Then
    Rechnen
    Speichern
  End If
  Initprocedure
End If
DataTab.MoveNext
Loop
DB.Close
End Function
```



Prozedur Initprocedure wird von Procedure Tippingcurve aufgerufen

Sub Initprocedure ()

```
Rem Begin Init
k = 1
Azimut = DataTab.Azimut
Elevation = DataTab.Elevation
quellsatznummer = DataTab.Datensatznummer
SA(1).Ga3 = (DataTab.[NB 31,5] - DataTab.[NA 31,5]) /
            (DataTab.[NH 31,5] -
             DataTab.[NA 31,5])
SA(1).Ga2 = (DataTab.[NB 23,8] - DataTab.[NA 23,8]) /
            (DataTab.[NH 23,8] -
             DataTab.[NA 23,8])
SA(1).m = 1 / Sin(DataTab.Elevation * PI / 180)
End Sub
```

Prozedur Rechnen wird von Procedure Tippingcurve aufgerufen

Sub Rechnen ()

```
Dim Sm, STB3, SmTB3, Sm2, STB2, SmTB2, STB32, STB22 As Double 'Summen der Regr.
Dim i As Integer
```

**Rem Begin Rechnen**

```
TH3 = 0: TH2 = 0: TA3 = 0: TA2 = 0
Sm = 0: Sm2 = 0
For i = 1 To k
    Sm = Sm + SA(i).m: Sm2 = Sm2 + (SA(i).m ^ 2)
    TA3 = TA3 + SA(i).TA3: TA2 = TA2 + SA(i).TA2
Next i
TA3 = TA3 / k: TA2 = TA2 / k
Do: Rem *** Berechnungen für Kanal 31,5 GHz ***
    STB3 = 0: SmTB3 = 0: STB32 = 0
    For i = 1 To k
        SA(i).TB3 = TA3 + (THI + TH3 - TA3) * SA(i).Ga3
        STB3 = STB3 + SA(i).TB3
        SmTB3 = SmTB3 + (SA(i).m * SA(i).TB3)
        STB32 = STB32 + (SA(i).TB3 ^ 2)
    Next i
    St3 = (SmTB3 - (Sm * STB3 / k)) / (Sm2 - (Sm ^ 2 / k))
    TB03 = (STB3 - (St3 * Sm)) / k
    TH3 = TH3 + ((TC - TB03) * (THI - TA3)) / (TB03 - TA3)
    Std3 = Sqr((STB32 - ((STB3 ^ 2) / k)) / (k - 1))
Loop Until Abs(TC - TB03) < .05
Do: Rem *** Berechnungen für Kanal 23,8 GHz ***
    STB2 = 0: SmTB2 = 0: STB22 = 0
    For i = 1 To k
        SA(i).TB2 = TA2 + (THI + TH2 - TA2) * SA(i).Ga2
        STB2 = STB2 + SA(i).TB2
        SmTB2 = SmTB2 + (SA(i).m * SA(i).TB2)
        STB22 = STB22 + (SA(i).TB2 ^ 2)
    Next i
    St2 = (SmTB2 - (Sm * STB2 / k)) / (Sm2 - (Sm ^ 2 / k))
    TB02 = (STB2 - (St2 * Sm)) / k
    TH2 = TH2 + ((TC - TB02) * (THI - TA2)) / (TB02 - TA2)
    Std2 = Sqr((STB22 - ((STB2 ^ 2) / k)) / (k - 1))
Loop Until Abs(TC - TB02) < .05
TH3 = THI + TH3: TH2 = THI + TH2: Rem THx = entsp. T(hot)
End Sub
```

### Prozedur Speichern wird von Procedure Tippingcurve aufgerufen

```

Sub Speichern ()

Dim i As Integer
Dim z As Variant

Rem Begin Speichern
ResultTab.AddNew
    ResultTab.TA3 = TA3
    ResultTab.TA2 = TA2
    ResultTab.[T(hot) 31,5] = TH3
    ResultTab.[T(hot) 23,8] = TH2
    ResultTab.[Standardabweichung 31,5] = Std3
    ResultTab.[Standardabweichung 23,8] = Std2
    ResultTab.Anzahlmessungen = k
ResultTab.Update
ResultTab.MoveLast
If (ResultTab.Indexnummer Mod 200) = 0 Then
    DoCmd Echo True, "Hot LoadTemp.Datensatznummer = "
        +ResultTab.Indexnummer
End If
For i = 0 To (k - 1)
    ConnectTab.AddNew
    ConnectTab.Datensatznummer = quellsatznummer + i
    ConnectTab.Indexnummer = ResultTab.Indexnummer
    ConnectTab.Update
Next i
End Sub

```

### Prozedur Link\_Meteo\_RadData verknüpft die Meteodatentabelle mit der Haupttabelle

```

Function Link_Meteo_RadData ()

Dim DB As Database
Dim MetTab As Dynaset: Dim DatTab As Table
Dim SteuerElement As Control
Dim low, high As Variant
Dim lowNr, highNr As Long
Dim GruppenName As String
Rem Begin Function
    Set DB = CurrentDB()
    Set SteuerElement = Forms![Hauptfenster]![AktData]
    GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
    DoCmd Hourglass True
    Set MetTab = DB.CreateDynaset(GruppenName + " Meteo")
    Set DatTab = DB.OpenTable(GruppenName)
    MetTab.Sort = "[Epoche]"
    DatTab.Index = "Epoche"
    MetTab.MoveFirst
    low = Date: lowNr = 0: high = low: highNr = 0
    If MetTab.EOF = False Then
        low = MetTab.[Epoche]
        lowNr = MetTab.[Datensatznummer]
    End If
    MetTab.MoveNext
    If MetTab.EOF = False Then
        high = MetTab.[Epoche]
        highNr = MetTab.[Datensatznummer]
    End If
End Function

```

```

DatTab.MoveFirst
Do While DatTab.EOF = False
  If Abs(DatTab.[Epoche] - low) > Abs(DatTab.[Epoche] - high) Then
    low = high
    lowNr = highNr
    MetTab.MoveNext
    If MetTab.EOF = False Then
      high = MetTab.[Epoche]
      highNr = MetTab.[Datensatznummer]
    End If
  End If
  DatTab.Edit
  DatTab.[MeteoNr] = lowNr
  DatTab.Update
  If (DatTab.Datensatznummer Mod 100) = 0 Then
    DoCmd Echo True, " Datensatznummer = " + DatTab.Datensatznummer
  End If
  DatTab.MoveNext
Loop
DoCmd Hourglass False
MetTab.Close
DatTab.Close
DB.Close
End Function

```

Prozedur `Tipping_LinReg` führt die lineare Regression über Heisslasttemperaturenmessreihe durch.

*'Diese Funktion führt eine lineare Regression über die einzelnen Messkampagnen aus. Ist die Messlücke grösser als eine Stunde wird ein neues Interval begonnen.*

```

Function Tipping_LinReg ()

Dim GN, SQL As String
Dim SteuerElement As Control
Dim Abfrage As QueryDef
Dim DatenAbfrage As Dynaset
Dim ResultTab As Table
Dim Startdatum, Enddatum, Zwischendatum As Variant
Dim SmX3, SmX2, SmX32, SmX22, SmY3, SmY2, SmY32, SmY22, SmXY3, SmXY2
As Double
Dim Steig As Double
Dim K2, k3 As Integer
Dim BeginHotLoad, EndHotLoad As Long

Rem Begin Function
DoCmd Hourglass True
Set DB = CurrentDB() ' Datenbank öffnen
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) ' Rem Gruppenname aus Formular übernehmen
Set Abfrage = DB.OpenQueryDef("Tipping LinReg")
Abfrage.SQL = "DELETE FROM [" + GN + " Tipping LinReg];"
Abfrage.Execute ' Alte Daten aus ... Tipping Linreg löschen
Set ResultTab = DB.OpenTable(GN + " Tipping LinReg")
SQL = "SELECT DISTINCTROW Avg([" + GN + "].Epoche) AS Epoche, "
SQL = SQL+"First([" + GN + " hot load temp].[Indexnummer]) AS
[Indexnummer], "
SQL = SQL+"First([" + GN + " hot load temp].[T(hot) 31,5]) AS
[T(hot) 31,5], "

```

```

SQL = SQL+"First([" + GN + " hot load temp].[T(hot) 23,8]) AS [T(hot) 23,8], "
SQL = SQL+"First([" +GN+ " hot load temp].[Standardabweichung 31,5])AS
      [StdAbw 31,5], "
SQL = SQL+"First([" +GN+ " hot load temp].[Standardabweichung 23,8]) AS
      [StdAbw 23,8] "
SQL = SQL+"FROM [" +GN+ "], [" +GN+ " Tipping to Data],
      [" + GN + " hot load temp],"
SQL = SQL+"[" + GN + " Tipping to Data] INNER JOIN ["+GN+" hot load
      temp] ON [" + GN + " Tipping to Data].Indexnummer=[" + GN
      + " hot load temp].Indexnummer, "
SQL = SQL + "[" + GN + " Tipping to Data] INNER JOIN [" +GN+ "] ON [" +
      GN+"Tipping to Data].Datensatznummer=["+GN+"].Datensatznummer "
SQL = SQL + "GROUP BY [" + GN + " hot load temp].Indexnummer "
'SQL = SQL + "ORDER BY Avg([" + GN + "].Epoche) "
SQL = SQL + "WITH OWNERACCESS OPTION; "
Set DatenAbfrage = DB.CreateDynaset(SQL) ' Daten für Berechnung in Dynaset
                                          ' DatenAbfrage aufbauen

DatenAbfrage.MoveFirst
DatenAbfrage.Sort = "[Epoche] Asc"
Do Until DatenAbfrage.EOF ' Berechnung durchführen
  Startdatum = DatenAbfrage.Epoche
  Enddatum = Startdatum
  Zwischendatum = Startdatum
  SmX3 = 0: SmX2 = 0: SmX32 = 0: SmX22 = 0
  SmY3 = 0: SmY32 = 0: SmY2 = 0
  SmY22 = 0: SmXY3 = 0: SmXY2 = 0: k3 = 0: K2 = 0
  BeginHotLoad = DatenAbfrage.Indexnummer
  Do Until ((DatenAbfrage.EOF) Or
            ((Zwischendatum - Enddatum) > MessLücke))
    If (DatenAbfrage.[StdAbw 31,5] < Std_Max_31) And
      (DatenAbfrage.[T(hot)
        31,5] > Min_31)And(DatenAbfrage.[T(hot) 31,5] < Max_31) Then
      SmX3 = SmX3 + (DatenAbfrage.Epoche - Int(Startdatum))
      SmX32 = SmX32 + ((DatenAbfrage.Epoche - Int(Startdatum)) ^ 2)
      SmY3 = SmY3 + DatenAbfrage.[T(hot) 31,5]
      SmY32 = SmY32 + (DatenAbfrage.[T(hot) 31,5] ^ 2)
      SmXY3 = SmXY3+((DatenAbfrage.Epoche-Int(Startdatum))*
        DatenAbfrage.[T(hot) 31,5])
      k3 = k3 + 1 ' Summen für lin. Reg. 31,5 GHz wurden aufgebaut
    End If
    If (DatenAbfrage.[StdAbw 23,8] < Std_Max_23) And
      (DatenAbfrage.[T(hot)
        23,8] > Min_23)And(DatenAbfrage.[T(hot) 23,8] < Max_23) Then
      SmX2 = SmX2 + (DatenAbfrage.Epoche - Int(Startdatum))
      SmX22 = SmX22 + ((DatenAbfrage.Epoche - Int(Startdatum)) ^ 2)
      SmY2 = SmY2 + DatenAbfrage.[T(hot) 23,8]: SmY22 = SmY22 +
        (DatenAbfrage.[T(hot) 23,8] ^ 2)
      SmXY2 = SmXY2 + ((DatenAbfrage.Epoche - Int(Startdatum)) *
        DatenAbfrage.[T(hot) 23,8])
      K2 = K2 + 1 ' Summen für lin. Reg. 23,8 GHz wurden aufgebaut
    End If
    Enddatum = DatenAbfrage.Epoche
    DatenAbfrage.MoveNext
    If Not DatenAbfrage.EOF Then
      Zwischendatum = DatenAbfrage.Epoche
      EndHotLoad = DatenAbfrage.Indexnummer
    End If
  Loop
  If (k3 > 1) And (K2 > 1) Then
    ResultTab.AddNew ' Regression berechnen und speichern
    ResultTab.BeginInterval = Startdatum
    ResultTab.Endinterval = Enddatum
    Steig = (SmXY3-((SmX3 * SmY3) / k3)) / (SmX32 - ((SmX3 ^ 2) / k3))
    ResultTab.[T(hot) 31,5 Ayx] = (SmY3 - (Steig * SmX3)) / k3
  End If
End Do

```

```

ResultTab.[T(hot) 31,5 Byx] = Steig
ResultTab.[T(hot) 31,5 Sy]=Sqr((SmY32-((SmY3 ^ 2) / k3))/(k3-1))
Steig = (SmXY2 - ((SmX2 * SmY2) / K2))/(SmX22-((SmX2 ^ 2) / K2))
ResultTab.[T(hot) 23,8 Ayx] = (SmY2 - (Steig * SmX2)) / K2
ResultTab.[T(hot) 23,8 Byx] = Steig
ResultTab.[T(hot) 23,8 Sy]=Sqr((SmY22-((SmY2 ^ 2)/K2))/(K2 - 1))
ResultTab.[HotLoadTempBegin] = BeginHotLoad
ResultTab.[HotLoadTempEnd] = EndHotLoad
ResultTab.Update
End If
Loop
DoCmd Hourglass False
End Function

```

Prozedur Pathdelay\_Calc berechnet den Pathdelay für die selektierte WVR-Datenreihe

```

Function Pathdelay_calc ()

Dim GN, SQL As String
Dim SteuerElement As Control
Dim Status As Integer
Dim TagBuff As Variant

Rem Begin Function Pathdelay_calc
DoCmd Hourglass True
Set DB = CurrentDB()           'Datenbank öffnen
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1)   'Rem Gruppename aus Formular übernehmen
Set DataTab = DB.OpenTable(GN)
Set LinregTab = DB.OpenTable(GN + " Tipping LinReg")
SQL = "SELECT DISTINCTROW Int([Epoche]) AS [Meteo Tag], "
SQL = SQL + "Max([" + GN + " Meteo].[Temperatur]) AS [TagesMax] "
SQL = SQL + "FROM [" + GN + " Meteo] "
SQL = SQL + "GROUP BY Int([Epoche]) "
SQL = SQL + "ORDER BY Int([Epoche]) "
SQL = SQL + "WITH OWNERACCESS OPTION; "
Set Tagesmax = DB.CreateDynaset(SQL)
DataTab.MoveFirst
LinregTab.Index = "PrimaryKey"
Tagesmax.MoveFirst
Do While DataTab.EOF = False
    Status = 2           'Passende Tipping-Kurve suchen
    If (DataTab.Epoche > LinregTab.BeginInterval) And (DataTab.Epoche <
        LinregTab.Endinterval) Then
        Status = 0           'Passende Tipping-Kurve gefunden
    Else
        If DataTab.Epoche < LinregTab.BeginInterval Then
            TagBuff = LinregTab.BeginInterval
            Do While (DataTab.Epoche < TagBuff) And (Not LinregTab.BOF)
                LinregTab.MovePrevious
                If Not LinregTab.BOF Then
                    TagBuff = LinregTab.BeginInterval
                End If
            Loop
        Else

```

```

TagBuff = LinregTab.Endinterval
Do While (DataTab.Epoche > TagBuff) And (Not LinregTab.EOF)
  LinregTab.MoveNext
  If Not LinregTab.EOF Then
    TagBuff = LinregTab.Endinterval
  End If
Loop
End If
If (LinregTab.EOF = True) Or (LinregTab.BOF = True) Then
  Status = 1      'Keine passende Tipping-Kurve gefunden'
  LinregTab.MoveFirst
Else
  If (DataTab.Epoche > LinregTab.BeginInterval) And (DataTab.Epoche <
    LinregTab.Endinterval) Then
    Status = 0    'Passende Tipping-Kurve gefunden'
  Else
    Status = 1    'Keine passende Tipping-Kurve gefunden'
  End If
End If
End If 'Ende (DataTab.Epoche > LinregDat....
If Status = 0 Then
  If Int(DataTab.Epoche) = Tagesmax.[Meteo Tag] Then
    Status = 0    'Passende Tagesmaximum gefunden'
  Else
    If (Int(DataTab.Epoche) < Tagesmax.[Meteo Tag]) Then
      TagBuff = Tagesmax.[Meteo Tag]
      Do While (Int(DataTab.Epoche) < TagBuff) And (Tagesmax.BOF=False)
        Tagesmax.MovePrevious
        If Not Tagesmax.BOF Then
          TagBuff = Tagesmax.[Meteo Tag]
        End If
      Loop
    Else
      TagBuff = Tagesmax.[Meteo Tag]
      Do While (Int(DataTab.Epoche) > TagBuff) And (Not Tagesmax.EOF)
        Tagesmax.MoveNext
        If Not Tagesmax.EOF Then
          TagBuff = Tagesmax.[Meteo Tag]
        End If
      Loop
    End If 'Ende (Int(DataTab.Epoche) < Tagesmax....
    If (Tagesmax.EOF = True) Or (Tagesmax.BOF = True) Then
      Status = 1      'Keine passende max. Temperatur gefunden'
      Tagesmax.MoveFirst
    Else
      If Int(DataTab.Epoche) = Tagesmax.[Meteo Tag] Then
        Status = 0    'Passende Tagesmaximum gefunden'
      Else
        Status = 1    'Keine passende max. Temperatur gefunden'
      End If 'Ende (Int(DataTab.Epoche) = Tagesmax....
    End If      'Ende (Tagesmax.EOF = ....'
  End If      'Ende Int(DataTab.Epoche = Tagesmax....'
End If      'Ende Status = 0 ....'
If Status = 0 Then
  DataTab.Edit
  DataTab.[Pathdelay calc] = True
  Rechnen_Pathdelay
  If (DataTab.Datensatznummer Mod 200) = 0 Then
    DoEvents
    DoCmd Echo True, " Datensatznummer = "+DataTab.Datensatznummer
  End If
  DataTab.Update
Else

```

```

    DataTab.Edit
    DataTab.[Pathdelay calc] = False
    DataTab.Update
End If
DataTab.MoveNext
Loop
DB.Close
DoCmd Hourglass False
End Function

```

Prozedur `Rechnen_Pathdelay` wird von der Prozedur `Pathdelay_calc` aufgerufen

```

Sub Rechnen_Pathdelay ()

Dim TA, TH, TB, Teff, Gamma As Double
Dim TB2, TB3, m As Double
Dim DeltaZeit As Variant

Rem Begin Sub
DeltaZeit = (DataTab.Epoche - Int(LinregTab.BeginInterval))
TA = (DataTab.[T(Horn) 31,5] + DataTab.[T(Switch) 31,5] +
      DataTab.[T(Atten)
      31,5] + DataTab.[T(Noise) 31,5]) / 4
Gamma = (DataTab.[NB 31,5] - DataTab.[NA 31,5]) / (DataTab.[NH 31,5] -
      DataTab.[NA 31,5])
TH = LinregTab.[T(hot) 31,5 Ayx] + (LinregTab.[T(hot) 31,5 Byx] *
      DeltaZeit)
TB = TA + ((TH - TA) * Gamma)
Teff = Ke * (Tagesmax.Tagesmax + TOC)
TB3 = TC - ((Teff - TC) * Log(Abs(1 - ((TB - TC) / (Teff - TC))))))
TA = (DataTab.[T(Horn) 23,8] + DataTab.[T(Switch) 23,8] +
      DataTab.[T(Atten)
      23,8] + DataTab.[T(Noise) 23,8]) / 4
Gamma = (DataTab.[NB 23,8] - DataTab.[NA 23,8]) / (DataTab.[NH 23,8] -
      DataTab.[NA 23,8])
TH = LinregTab.[T(hot) 23,8 Ayx] + (LinregTab.[T(hot) 23,8 Byx] *
      DeltaZeit)
TB = TA + ((TH - TA) * Gamma)
TB2 = TC - ((Teff - TC) * Log(Abs(1 - ((TB - TC) / (Teff - TC))))))
m = 1 / Sin((DataTab.Elevation) * PI / 180)
DataTab.Pathdelay = A0 * m + A1 * TB2 + A2 * TB3
End Sub

```

Prozedur `Tipping_linReg_Nachbearb` führt eine lineare Regression über die manuell eingegebenen Zeitintervalle durch.

*'Diese Funktion führt eine Nachbearbeitung der lineare Regression über die einzelnen Messkampagnen aus.*

```

Function Tipping_LinReg_Nachbearb ()

Dim GN, SQL As String
Dim SteuerElement As Control
Dim Abfrage As QueryDef
Dim DatenAbfrage As Dynaset
Dim LinRegAbfrage As Dynaset
Dim ResultTab As Table
Dim HotLoadTab As Table
Dim Startdatum, Enddatum, Zwischendatum As Variant
Dim SmX3, SmX2, SmX32, SmX22, SmY3, SmY2, SmY32, SmY22, SmXY3, SmXY2 As Double
Dim Steig, T31, T23 As Double
Dim K2, k3 As Integer

```

```
Dim BeginHotLoad, EndHotLoad, StartIndex, EndIndex, AktIndex As Long
Dim Krit As String
```

```
Rem Begin Function
```

```
Set DB = CurrentDB() ' Datenbank öffnen
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) ' Rem Gruppenname aus Formular übernehmen
*** Lin. Reg für alle Intervale in ... Tipping LinReg neu berechnen ***
Set ResultTab = DB.OpenTable(GN + " Tipping LinReg")
SQL = "SELECT DISTINCTROW Avg([" + GN + "].Epoche) AS Epoche, "
SQL = SQL + "First([" + GN + " hot load temp].[Indexnummer]) AS "
SQL = SQL + " [Indexnummer], "
SQL = SQL + "First([" + GN + " hot load temp].[T(hot) 31,5]) AS [T(hot) "
SQL = SQL + " 31,5], "
SQL = SQL + "First([" + GN + " hot load temp].[T(hot) 23,8]) AS [T(hot) "
SQL = SQL + " 23,8], "
SQL = SQL + "First([" + GN + " hot load temp].[Standardabweichung 31,5]) AS "
SQL = SQL + " [StdAbw 31,5], "
SQL = SQL + "First([" + GN + " hot load temp].[Standardabweichung 23,8]) AS "
SQL = SQL + " [StdAbw 23,8], "
SQL = SQL + "First([" + GN + " hot load temp].[Status 31,5]) AS [Status "
SQL = SQL + " 31,5], "
SQL = SQL + "First([" + GN + " hot load temp].[Status 23,8]) AS [Status "
SQL = SQL + " 23,8] "
SQL = SQL + "FROM [" + GN + "], [" + GN + " Tipping to Data], [" + GN + " "
SQL = SQL + " hot load temp], "
SQL = SQL + "[" + GN + " Tipping to Data] INNER JOIN [" + GN + " hot load "
SQL = SQL + " temp] ON [" + GN + " Tipping to Data].Indexnummer = [" + GN + " "
SQL = SQL + " hot load temp].Indexnummer, "
SQL = SQL + "[" + GN + " Tipping to Data] INNER JOIN [" + GN + "] ON [" + "
SQL = SQL + " GN + " Tipping to Data].Datensatznummer = [" + GN + " "
SQL = SQL + "].Datensatznummer "
SQL = SQL + "GROUP BY [" + GN + " hot load temp].Indexnummer "
SQL = SQL + "ORDER BY Avg([" + GN + "].Epoche) "
SQL = SQL + "WITH OWNERACCESS OPTION; "
Set DatenAbfrage = DB.CreateDynaset(SQL) ' Daten für Berechnung in Dynaset
' DatenAbfrage aufbauen

DoEvents
ResultTab.MoveFirst
DatenAbfrage.MoveFirst
Do Until ResultTab.EOF
Startdatum = ResultTab.BeginInterval
Enddatum = ResultTab.Endinterval
Krit = "[Epoche] >= " + Format(Startdatum, "0.00000000")
DatenAbfrage.FindFirst Krit
If Not DatenAbfrage.NoMatch Then
SmX3 = 0: SmX2 = 0: SmX32 = 0: SmX22 = 0: SmY3 = 0: SmY32 = 0
SmY2 = 0: SmY22 = 0: SmXY3 = 0: SmXY2 = 0: k3 = 0: K2 = 0
BeginHotLoad = DatenAbfrage.Indexnummer
Zwischendatum = DatenAbfrage.[Epoche]
Do Until ((DatenAbfrage.EOF) Or (Format(Enddatum - Zwischendatum,
"0.00000000") <= 0))
If (DatenAbfrage.[StdAbw 31,5] < Std_Max_31) And
(DatenAbfrage.[T(hot) 31,5] > Min_31) And (DatenAbfrage.[T(hot)
31,5] < Max_31) Then
SmX3 = SmX3 + (DatenAbfrage.Epoche - Int(Startdatum))
SmX32 = SmX32 + ((DatenAbfrage.Epoche - Int(Startdatum)) ^ 2)
SmY3 = SmY3 + DatenAbfrage.[T(hot) 31,5]
SmY32 = SmY32 + (DatenAbfrage.[T(hot) 31,5] ^ 2)
SmXY3 = SmXY3 + ((DatenAbfrage.Epoche - Int(Startdatum)) *
DatenAbfrage.[T(hot) 31,5])

k3 = k3 + 1 ' Summen für lin. Reg. 31,5 GHz wurden aufgebaut
End If
If (DatenAbfrage.[StdAbw 23,8] < Std_Max_23) And
```



```

    (DatenAbfrage.[T(hot) 23,8]>Min_23)And(DatenAbfrage.[T(hot)
      23,8] < Max_23) Then
    SmX2 = SmX2 + (DatenAbfrage.Epoche - Int(Startdatum))
    SmX22 = SmX22 + ((DatenAbfrage.Epoche-Int(Startdatum)) ^ 2)
    SmY2 = SmY2 + DatenAbfrage.[T(hot) 23,8]
    SmY22 = SmY22 + (DatenAbfrage.[T(hot) 23,8] ^ 2)
    SmXY2 = SmXY2 + ((DatenAbfrage.Epoche - Int(Startdatum)) *
      DatenAbfrage.[T(hot) 23,8])
    K2 = K2 + 1
      ' Summen für lin. Reg. 23,8 GHz wurden aufgebaut
End If
DoCmd Echo True, " Aktuelles Datum = " +
  Format(DatenAbfrage.[Epoche], "dddd tttt")
DatenAbfrage.MoveNext
If Not DatenAbfrage.EOF Then
  Zwischendatum = DatenAbfrage.Epoche
  EndHotLoad = DatenAbfrage.Indexnummer
End If
Loop
If (k3 > 1) And (K2 > 1) Then
  ResultTab.Edit
    ' Regression berechnen und speichern
  Steig = (SmXY3-((SmX3 * SmY3)/k3))/((SmX32 - ((SmX3 ^ 2)/k3))
  ResultTab.[T(hot) 31,5 Ayx] = (SmY3 - (Steig * SmX3)) / k3
  ResultTab.[T(hot) 31,5 Byx] = Steig
  ResultTab.[T(hot) 31,5 Sy]=Sqr((SmY32-((SmY3^2)/k3))/(k3-1))
  Steig = (SmXY2-((SmX2 * SmY2)/K2))/((SmX22 - ((SmX2 ^ 2)/K2))
  ResultTab.[T(hot) 23,8 Ayx] = (SmY2 - (Steig * SmX2)) / K2
  ResultTab.[T(hot) 23,8 Byx] = Steig
  ResultTab.[T(hot) 23,8 Sy]=Sqr((SmY22-((SmY2^2)/K2))/(K2-1))
  ResultTab.[HotLoadTempBegin] = BeginHotLoad
  ResultTab.[HotLoadTempEnd] = EndHotLoad
  ResultTab.Update
End If
End If
DoEvents
ResultTab.MoveNext
Loop

*** Hot load Temp. markieren ***
Set LinRegAbfrage = DB.CreateDynaset(GN + " Tipping LinReg")
LinRegAbfrage.Sort = "[BeginInterval]"
Set HotLoadTab = DB.OpenTable(GN + " hot load temp")
HotLoadTab.Index = "PrimaryKey"
HotLoadTab.MoveFirst
LinRegAbfrage.MoveFirst
Do Until LinRegAbfrage.EOF
  StartIndex = LinRegAbfrage.HotLoadTempBegin
  EndIndex = LinRegAbfrage.HotLoadTempEnd
  Krit = "[Indexnummer] = " + StartIndex
  DatenAbfrage.FindFirst Krit
  AktIndex = DatenAbfrage.Indexnummer
  Do Until AktIndex > EndIndex
    T31 =LinRegAbfrage.[T(hot) 31,5 Ayx]+(LinRegAbfrage.[T(hot) 31,5
      Byx] * (DatenAbfrage.Epoche - Int(DatenAbfrage.Epoche)))
    T23 =LinRegAbfrage.[T(hot) 23,8 Ayx]+(LinRegAbfrage.[T(hot) 23,8
      Byx] * (DatenAbfrage.Epoche - Int(DatenAbfrage.Epoche)))
    HotLoadTab.Seek "=", AktIndex
    DoCmd Echo True, " Aktueller Index = "+DatenAbfrage.Indexnummer
    HotLoadTab.Edit
    If(DatenAbfrage.[T(hot) 31,5]<(T31+(2*LinRegAbfrage.[T(hot) 31,5
      Sy])) And (DatenAbfrage.[T(hot) 31,5] > (T31 - (2 *
      LinRegAbfrage.[T(hot) 31,5 Sy])))) Then
      HotLoadTab.[Status 31,5] = True
    Else
      HotLoadTab.[Status 31,5] = False
  
```

```

End If
If (DatenAbfrage.[T(hot) 23,8] < (T23 + (2 * LinRegAbfrage.[T(hot) 23,8
  Sy]))) And (DatenAbfrage.[T(hot) 23,8] > (T23 - (2 *
  LinRegAbfrage.[T(hot) 23,8 Sy]))) Then
  HotLoadTab.[Status 23,8] = True
Else
  HotLoadTab.[Status 23,8] = False
End If
HotLoadTab.Update
DatenAbfrage.MoveNext
If Not DatenAbfrage.EOF Then
  AktIndex = DatenAbfrage.Indexnummer
Else
  AktIndex = AktIndex + 10000
End If
Loop
DoEvents
LinRegAbfrage.MoveNext
Loop
*** lineare Regression mit bereinigten Daten ***
ResultTab.MoveFirst
DatenAbfrage.MoveFirst
Do Until ResultTab.EOF
  Startdatum = ResultTab.BeginInterval
  Enddatum = ResultTab.Endinterval
  Krit = "[Epoche] >= " + Format(Startdatum, "0.00000000")
  DatenAbfrage.FindFirst Krit
  If Not DatenAbfrage.NoMatch Then
    SmX3 = 0: SmX2 = 0: SmX32 = 0: SmX22 = 0: SmY3 = 0: SmY32 = 0
    SmY2 = 0: SmY22 = 0: SmXY3 = 0: SmXY2 = 0: k3 = 0: K2 = 0
    BeginHotLoad = DatenAbfrage.Indexnummer
    Zwischendatum = DatenAbfrage.[Epoche]
    Do Until ((DatenAbfrage.EOF) Or (Format(Enddatum - Zwischendatum,
      "0.00000000") <= 0))
      If DatenAbfrage.[Status 31,5] = True Then
        SmX3 = SmX3 + (DatenAbfrage.Epoche - Int(Startdatum))
        SmX32 = SmX32 + ((DatenAbfrage.Epoche - Int(Startdatum)) ^ 2)
        SmY3 = SmY3 + DatenAbfrage.[T(hot) 31,5]
        SmY32 = SmY32 + (DatenAbfrage.[T(hot) 31,5] ^ 2)
        SmXY3 = SmXY3 + ((DatenAbfrage.Epoche - Int(Startdatum)) *
          DatenAbfrage.[T(hot) 31,5])
        k3 = k3 + 1
        ' Summen für lin. Reg. 31,5 GHz wurden aufgebaut
      End If
      If DatenAbfrage.[Status 23,8] = True Then
        SmX2 = SmX2 + (DatenAbfrage.Epoche - Int(Startdatum)): SmX22 =
          SmX22 + ((DatenAbfrage.Epoche - Int(Startdatum)) ^ 2)
        SmY2 = SmY2 + DatenAbfrage.[T(hot) 23,8]: SmY22 = SmY22 +
          (DatenAbfrage.[T(hot) 23,8] ^ 2)
        SmXY2 = SmXY2 + ((DatenAbfrage.Epoche - Int(Startdatum)) *
          DatenAbfrage.[T(hot) 23,8])
        K2 = K2 + 1
        ' Summen für lin. Reg. 23,8 GHz wurden aufgebaut
      End If
    DoCmd Echo True, "Aktuelles Datum=" + Format(DatenAbfrage.[Epoche],
      "dddd tttt")
    DatenAbfrage.MoveNext
    If Not DatenAbfrage.EOF Then
      Zwischendatum = DatenAbfrage.Epoche
      EndHotLoad = DatenAbfrage.Indexnummer
    End If
  Loop
  If (k3 > 1) And (K2 > 1) Then
    ResultTab.Edit
    Steig = (SmXY3 - ((SmX3 * SmY3) / k3)) / (SmX32 - ((SmX3 ^ 2) / k3))
    ResultTab.[T(hot) 31,5 Ayx] = (SmY3 - (Steig * SmX3)) / k3
  End If

```

```

ResultTab.[T(hot) 31,5 Byx] = Steig
ResultTab.[T(hot) 31,5 Sy]=Sqr((SmY32-((SmY3^2)/k3))/(k3-1))
Steig = (SmXY2 - ((SmX2 * SmY2)/K2))/(SmX22-((SmX2 ^ 2)/K2))
ResultTab.[T(hot) 23,8 Ayx] = (SmY2 - (Steig * SmX2)) / K2
ResultTab.[T(hot) 23,8 Byx] = Steig
ResultTab.[T(hot) 23,8 Sy]=Sqr((SmY22-((SmY2^2)/K2))/(K2-1))
ResultTab.[HotLoadTempBegin] = BeginHotLoad
ResultTab.[HotLoadTempEnd] = EndHotLoad
ResultTab.Update
End If
End If
DoEvents
ResultTab.MoveNext
Loop
End Function

```

Prozedur Pathdelay\_calc\_ohnereg berechnet den Path Delay ohne eine lineare Regression über die Heisslasttemperaturmessreihe.

```

Function Pathdelay_calc_ohnereg ()

Dim GN, SQL As String
Dim SteuerElement As Control
Dim Status As Integer
Dim TagBuff As Variant
Dim DataSet As Dynaset
Dim TA, TH, TB, Teff, Gamma As Double
Dim TB2, TB3, m As Double

Rem Begin Function Pathdelay_calc
DoCmd Hourglass True
Set DB = CurrentDB() ' Datenbank öffnen
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) ' Rem Gruppename aus Formular übernehmen
Set DataTab = DB.OpenTable(GN)
SQL = "SELECT DISTINCTROW [" + GN + "].*, [" + GN + " Tipping to Data].*, "
SQL = SQL + "[" + GN + " hot load temp].* "
SQL = SQL + "FROM [" + GN + "], [" + GN + " Tipping to Data], [" + GN + " "
SQL = SQL + "hot load temp],"
SQL = SQL + "[" + GN + "] INNER JOIN [" + GN + " Tipping to Data] ON [" + GN + "].Datensatznummer = [" + GN + " Tipping to Data].Datensatznummer,"
SQL = SQL + "[" + GN + " Tipping to Data] INNER JOIN [" + GN + " hot load temp] ON [" + GN + " Tipping to Data].Indexnummer = [" + GN + " hot load temp].Indexnummer"
SQL = SQL + " WITH OWNERACCESS OPTION;"
Set DataSet = DB.CreateDynaset(SQL)
SQL = "SELECT DISTINCTROW Int([Epoche]) AS [Meteo Tag], "
SQL = SQL + "Max([" + GN + " Meteo].[Temperatur]) AS [TagesMax] "
SQL = SQL + "FROM [" + GN + " Meteo] "
SQL = SQL + "GROUP BY Int([Epoche]) "
SQL = SQL + "ORDER BY Int([Epoche]) "
SQL = SQL + "WITH OWNERACCESS OPTION;"
Set Tagesmax = DB.CreateDynaset(SQL)
DataSet.MoveFirst
DataTab.MoveFirst
Tagesmax.MoveFirst
Do While DataSet.EOF = False
If (DataSet.Epoche) = Tagesmax.[Meteo Tag] Then
Status = 0 ' Passende Tagesmaximum gefunden
Else
If (Int(DataSet.Epoche) < Tagesmax.[Meteo Tag]) Then
TagBuff = Tagesmax.[Meteo Tag]

```

```

Do While(Int(DataSet.Epoche)<TagBuff)And(Tagesmax.BOF = False)
  Tagesmax.MovePrevious
  If Not Tagesmax.BOF Then
    TagBuff = Tagesmax.[Meteo Tag]
  End If
Loop
Else
  TagBuff = Tagesmax.[Meteo Tag]
  Do While(Int(DataSet.Epoche)>TagBuff) And (Not Tagesmax.EOF)
    Tagesmax.MoveNext
    If Not Tagesmax.EOF Then
      TagBuff = Tagesmax.[Meteo Tag]
    End If
  Loop
End If 'Ende (Int(DataSet.Epoche) < Tagesmax....
If (Tagesmax.EOF = True) Or (Tagesmax.BOF = True) Then
  Status = 1 ' Keine passende max. Temperatur gefunden
  Tagesmax.MoveFirst
Else
  If Int(DataSet.Epoche) = Tagesmax.[Meteo Tag] Then
    Status = 0 ' Passende Tagesmaximum gefunden
  Else
    Status = 1 ' Keine passende max. Temperatur gefunden
  End If 'Ende (Int(DataSet.Epoche) = Tagesmax....
End If 'Ende (Tagesmax.EOF = ....
End If 'Ende Int(DataSet.Epoche = Tagesmax....
If Status = 0 Then
  DataTab.Edit
  DataTab.[Pathdelay calc] = True
  TA = (DataSet.[T(Horn) 31,5] + DataSet.[T(Switch) 31,5] +
    DataSet.[T(Atten) 31,5] + DataSet.[T(Noise) 31,5]) / 4
  Gamma=(DataSet.[NB 31,5]-DataSet.[NA 31,5])/(DataSet.[NH 31,5]
    - DataSet.[NA 31,5])
  TH = DataSet.[T(hot) 31,5]
  TB = TA + ((TH - TA) * Gamma)
  Teff = Ke * (Tagesmax.Tagesmax + T0C)
  TB3 = TC-((Teff-TC) * Log(Abs(1 - ((TB - TC) / (Teff - TC))))))
  TA = (DataSet.[T(Horn) 23,8] + DataSet.[T(Switch) 23,8] +
    DataSet.[T(Atten) 23,8] + DataSet.[T(Noise) 23,8]) / 4
  Gamma=(DataSet.[NB 23,8]-DataSet.[NA 23,8])/(DataSet.[NH 23,8]
    - DataSet.[NA 23,8])
  TH = DataSet.[T(hot) 23,8]
  TB = TA + ((TH - TA) * Gamma)
  TB2 = TC-((Teff - TC)*Log(Abs(1 - ((TB - TC) / (Teff - TC))))))
  m = 1 / Sin((DataSet.Elevation - 5) * PI / 180)
  DataTab.Pathdelay = A0 * m + A1 * TB2 + A2 * TB3
  If (DataTab.Datensatznummer Mod 200) = 0 Then
    DoEvents
    DoCmd Echo True, " Datensatznummer = "+DataTab.Datensatznummer
  End If
  DataTab.Update
Else
  DataTab.Edit
  DataTab.[Pathdelay calc] = False
  DataTab.Update
End If
DataSet.MoveNext
DataTab.MoveNext
Loop
DB.Close
DoCmd Hourglass False
End Function

```

## Anhang E

### Makro Anzeigen

Um die Prozeduren des Moduls Einlesen ausführen zu können, müssen zuerst die entsprechenden Makros aufgerufen werden.

Makronamen : Anzeigen			
Makrogruppe	Aktion	Bedingung	Argument1
Messdaten	AusführenCode		Messdaten_Darstellung()
TippUndData	AusführenCode		Tippingcurve_Darstellung()
MeteoDaten	AusführenCode		Meteodaten_Darstellen()
TippingDatum	AusführenCode		Tippingcurve_Datum()
UnterFormAkt	AusführenCode		Tippingcurve_Datum_UnterForm()
TippingOrigin	AusführenCode		Tipping_to_Origin()
UnterFormOrigin	AusführenCode		Tipping_to_Origin_Update()
PathDatum	AusführenCode		Pathdelay_Diagramm()
UnterformPath	AusführenCode		Pathdelay_Diagramm_Unterform()
PathSmooth	AusführenCode		Pathdelay_Smooth()
UnterformSmooth	AusführenCode		Pathdelay_Smooth_Unterform()

*Tabelle 8 : Makrogruppe Anzeigen*

### Deklarationen im Modul Anzeigen

Option Compare Database    *'Verwenden der Datenbank-Sortierreihenfolge beim Vergleich von Zeichenfolgen.*

Option Explicit

```
Dim DB As Database
Dim DatenTab As Dynaset
Dim Abfragedatum As Dynaset
Dim FormFilter As Dynaset
Dim Abfrage As QueryDef
Dim AbfrageSelect As QueryDef
```

```
Dim inf As Variant
```

```
Const AnzahlPunkte = 5
Const AnzahlIter = 1
Const MaxWert = 20
Const MinWert = 2
```

### Prozedur Messdaten\_Darstellen

```

Function Messdaten_Darstellung ()

Dim DB As Database
Dim Abfrage As QueryDef
Dim SteuerElement As Control
Dim GruppenName As String

Rem Begin Function
  Set DB = CurrentDB()
  Set SteuerElement = Forms![Hauptfenster]![AktData]
  GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
  Set Abfrage = DB.CreateQueryDef("RohDaten",
    "Select * From [" + GruppenName + "];")

  DoCmd OpenForm "Messdaten", A_Normal, , , A_Edit, A_Normal
  Abfrage.Close
  DB.DeleteQueryDef ("RohDaten")
  DB.Close
End Function

```

### Prozedur Tippingcurve\_Darstellen

```

Function Tippingcurve_Darstellung ()

Dim DB As Database
Dim Abfrage As QueryDef: Dim UnterAbfrage As QueryDef
Dim SteuerElement As Control
Dim GruppenName, SQLText As String

Rem Begin Function
  Set DB = CurrentDB()
  Set SteuerElement = Forms![Hauptfenster]![AktData]
  GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
  SQLText = "SELECT DISTINCTROW [" + GruppenName + " Tipping to
  Data].Indexnummer, Epoche, Elevation, Azimut, [NA 31,5], [NH 31,5],
  [NB 31,5], [NA 23,8], [NH 23,8], [NB 23,8] FROM [" + GruppenName + "
  Tipping to Data], [" + GruppenName + "], [" + GruppenName + " Tipping
  to Data] INNER JOIN [" + GruppenName + "] ON [" + GruppenName + "
  Tipping to Data].Datensatznummer = [" + GruppenName + "
  "].Datensatznummer ;"
  Set UnterAbfrage = DB.CreateQueryDef("TipUnterDaten", SQLText)
  SQLText = "SELECT DISTINCT [" + GruppenName + " hot load
  temp].Indexnummer,TA3,TA2,Azimut,[T(hot) 31,5],[T(hot) 23,8],
  [Standardabweichung 31,5],[Standardabweichung 23,8], Anzahlmessungen
  FROM [" + GruppenName + "], [" + GruppenName + " hot load temp], [" +
  GruppenName + " Tipping to Data], [" + GruppenName + " Tipping to
  Data] INNER JOIN [" + GruppenName + " hot load temp] ON [" +
  GruppenName + " Tipping to Data].Indexnummer = [" + GruppenName + "
  hot load temp].Indexnummer,[" + GruppenName + " Tipping to Data] INNER
  JOIN [" + GruppenName + "] ON [" + GruppenName + " Tipping to
  Data].Datensatznummer = [" + GruppenName + "].Datensatznummer ORDER BY
  [" + GruppenName + " hot load temp].Indexnummer ;"
  Set Abfrage = DB.CreateQueryDef("TipDaten", SQLText)
  DoCmd OpenForm "Tippingdaten", A_Normal, , , A_Edit, A_Normal
  UnterAbfrage.Close
  Abfrage.Close
  DB.DeleteQueryDef ("TipUnterDaten")
  DB.DeleteQueryDef ("TipDaten")
  DB.Close
End Function

```

## Prozedur MeteoDaten\_Darstellen

```
Function MeteoDaten_Darstellen ()

Dim DB As Database
Dim Abfrage As QueryDef
Dim SteuerElement As Control
Dim GruppenName As String

Rem Begin Function
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GruppenName=SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
Rem DB.DeleteQueryDef ("MeteoDaten")
Set Abfrage = DB.OpenQueryDef("MeteoDaten")
Abfrage.SQL = "Select * From [" + GruppenName + " Meteo];"
DoCmd OpenForm "Meteodaten", A_Normal, , , A_Edit, A_Normal
Abfrage.Close
DB.Close
End Function
```

## Prozedur Tipping\_Datum

*'Diese Procedure baut die Tabelle TippingGraf mit den Daten für  
'die graphische Darstellung der Tipping-Kurve auf. Ist die ausgewählte  
'Datengruppe dieselbe wie die vorhande, kann eine neue oder die be-  
'reits vorhande Tabelle verwendet werden*

```
Function Tippingcurve_Datum ()

Dim SteuerTag As Control
Dim SteuerElement As Control
Dim GN, SQL, Krit As String
Dim Status As Integer
Dim Graf As Table
Dim Linreg As Dynaset
Dim StartIndex, EndIndex As Long
Dim DeltaZeit As Variant

Rem Begin Function
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) *** GruppenName aus Formular übernehmen
Set Abfrage = DB.OpenQueryDef("TippingGraf Create")
Rem *** suchen ob Tabellen und aktuelle Datensatzgruppe übereinstimmen
If InStr(60, Abfrage.SQL, GN, 0) = 0 Then
    Status = True
Else
    If MsgBox("Möchten Sie die Datensätze zu den Diagrammen aktualisieren  
?", 292, "Smooth aktualisieren") = 6 Then
        Status = True
    Else
        Status = False
    End If
End If
If Status = True Then
    DoCmd Hourglass True
    DoCmd Echo True, "Alte Daten der Tipping-Kurve-Diagramme löschen"
    Abfrage.SQL = "DELETE FROM TippingGraf;"
    Abfrage.Execute
    DoCmd Echo True, "Datentabelle für Tipping-Kurve-Diagramm wird  
erstellt"
    SQL = "Insert INTO TippingGraf (DataZeit,[T(hot) 31,5],[T(hot)  
23,8],Indexnummer,Epoche) "
```

```

SQL = SQL + "SELECT DISTINCTROW (Avg(["+GN + "].Epoche)-Int(Avg(["
      + GN + "].Epoche)))*24 As DataZeit,"
SQL = SQL + "First([" + GN + " hot load temp].[T(hot) 31,5])as
      [T(hot) 31,5],"
SQL = SQL + "First([" + GN + " hot load temp].[T(hot) 23,8])as
      [T(hot) 23,8],"
SQL = SQL + "First([" + GN + " hot load temp].[Indexnummer])as
      [Indexnummer],"
SQL = SQL + "Int(Avg([" + GN + "].Epoche))as[Epoche]"
SQL = SQL + "FROM [" + GN + "], [" + GN + " hot load temp],
      [" + GN + " Tipping to Data], "
SQL = SQL + "[" + GN + "] RIGHT JOIN ["+GN + " Tipping to Data] ON
      [" + GN + "].Datensatznummer = [" + GN +
      " Tipping to Data].Datensatznummer,"
SQL = SQL + "[" + GN + " hot load temp] INNER JOIN [" + GN +
      " Tipping to Data] ON ["+GN + " hot load temp].Indexnummer
      = [" + GN + " Tipping to Data].Indexnummer "
SQL = SQL + " Group BY [" + GN + " Tipping to Data].Indexnummer "
SQL = SQL + " ORDER BY Int(Avg(["+GN + "].Epoche)), (Avg([" + GN +
      "].Epoche)-Int(Avg([" + GN + "].Epoche)))*24;"

Abfrage.SQL = SQL
Abfrage.Execute

*** lin. Regressionsgerade in Anzeigefile kopieren ***
Set Linreg = DB.CreateDynaset(GN + " Tipping LinReg")
Linreg.Sort = "[BeginInterval]"
Set Graf = DB.OpenTable("TippingGraf")
Graf.MoveFirst
Linreg.MoveFirst
StartIndex = Linreg.HotLoadTempBegin
EndIndex = Linreg.HotLoadTempEnd - 1
Do Until Graf.EOF
  If (Graf.Indexnummer >= StartIndex) And
    (Graf.Indexnummer <= EndIndex) Then
    DeltaZeit = Graf.DataZeit / 24 +
      (Graf.Epoche - Format(Int(Linreg.[BeginInterval]), "0"))
    Graf.Edit
    Graf.[lin T(hot) 31,5] = Linreg.[T(hot) 31,5 Ayx] +
      (DeltaZeit * Linreg.[T(hot) 31,5 Byx])
    Graf.[lin T(hot) 23,8] = Linreg.[T(hot) 23,8 Ayx] +
      (DeltaZeit * Linreg.[T(hot) 23,8 Byx])
    Graf.Update
  Else
    Krit = "[HotLoadTempEnd] > " + Graf.Indexnummer
    Linreg.FindFirst Krit
    StartIndex = Linreg.HotLoadTempBegin
    EndIndex = Linreg.HotLoadTempEnd
  End If
  Graf.MoveNext
Loop
DoCmd Hourglass False
End If
DoCmd Echo True, "Datumsliste wird erstellt"
SQL = "SELECT DISTINCT [Epoche] As Datum FROM [TippingGraf]
      ORDER BY [Epoche];"
Set Abfragedatum = DB.CreateDynaset(SQL)
DoCmd OpenForm "Tipping-Kurve/Diag nach Tagen",A_Normal, , ,A_Edit,A_Normal
Abfragedatum.MoveFirst
Forms![Tippingcurve/Diag nach Tagen]![GruppenBezeichnung] = GN
Forms![Tippingcurve/Diag nach Tagen]![DatumSelect] =
  Format(Abfragedatum.Datum, "dd.mm.yyyy")
SQL ="SELECT DISTINCTROW Format(DataZeit,'00.00') as Tippingcurve, "
SQL = SQL + "[T(hot) 31,5],"
SQL = SQL + "[T(hot) 23,8],"
SQL = SQL + "[lin T(hot) 31,5],"

```



```

SQL = SQL + "[lin T(hot) 23,8] "
SQL = SQL + "FROM TippingGraf "
SQL = SQL + "WHERE [Epoche] = " + Str$(Abfragedatum.Datum) + " "
SQL = SQL + "ORDER BY DataZeit ;"
Set AbfrageSelect = DB.OpenQueryDef("TippingGraf Select")
AbfrageSelect.SQL = SQL
End Function

```

Prozedur Tippingcurve\_Datum\_UnterForm wird vom Makro UnterFormAkt aufgerufen.

```

Function Tippingcurve_Datum_UnterForm ()

Dim DB As Database
Dim SteuerElement As Control
Dim GruppenName, SQLText As String
Dim ddddd As Long

Rem Begin Function
dddd = DateSerial(Year(Forms![Tippingcurve/Diag nach
Tagen]![DatumSelect]), Month(Forms![Tippingcurve/Diag nach
Tagen]![DatumSelect]), Day(Forms![Tippingcurve/Diag nach
Tagen]![DatumSelect]))
SQLText="SELECT DISTINCTROW Format(DataZeit,'00.00') as Tippingcurve, "
SQLText = SQLText + "[T(hot) 31,5], "
SQLText = SQLText + "[T(hot) 23,8], "
SQLText = SQLText + "[lin T(hot) 31,5], "
SQLText = SQLText + "[lin T(hot) 23,8] "
SQLText = SQLText + "FROM TippingGraf "
SQLText = SQLText + "WHERE [Epoche]= " + Str$(dddd) + " "
SQLText = SQLText + "ORDER BY DataZeit ;"
AbfrageSelect.SQL = SQLText
DoCmd Requery "Tipping-Diagramm"
End Function

```

Prozedur Tipping\_Interval dient zur Festlegung der Zeitintervalle einer Messreihe

```

Function Tipping_Interval ()

Dim DB As Database
Dim Abfrage As QueryDef
Dim SteuerElement As Control
Dim GruppenName As String

Rem Begin Function
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GruppenName = SteuerElement.Spalte(1) 'Gruppenname aus Formular übernehmen
Set Abfrage = DB.CreateQueryDef("IntervalDaten", "Select
BeginInterval,Endinterval From ["+GruppenName+" Tipping LinReg];")
DoCmd OpenForm "TippingInterval", A_Normal, , , A_Edit, A_Normal
Abfrage.Close
DB.DeleteQueryDef ("IntervalDaten")
DB.Close
End Function

```

## Prozedur Pathdelay\_Diagramm

*'Diese Procedure baut die Abfrage mit den Daten für die graphische Darstellung des Pathdelays auf.*

```
Function Pathdelay_Diagramm ()
```

```
Dim SteuerTag As Control
Dim SteuerElement As Control
Dim GN, SQL As String
Dim Status As Integer
Dim ddddd As Long
```

**Rem Begin Function**

```
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) *** GruppenName aus Formular übernehmen
Set Abfrage = DB.OpenQueryDef("Pathdelay Create")
Rem *** suchen ob Tabellen und aktuelle Datensatzgruppe übereinstimmen
If InStr(60, Abfrage.SQL, GN, 0) = 0 Then
    Status = True
Else
    If MsgBox("Möchten Sie die Datensätze zu den Diagrammen aktualisieren?", 292, "Diagramm aktualisieren") = 6 Then
        Status = True
    Else
        Status = False
    End If
End If
If Status = True Then
    DoCmd Hourglass True
    DoCmd Echo True, "Alte Daten der Pathdelay-Diagramme löschen"
    Abfrage.SQL = "DELETE FROM [Pathdelay-Diagramm] ;"
    Abfrage.Execute
    DoCmd Echo True, "Datentabelle für Pathdelay-Diagramme wird erstellt"
    SQL = "Insert INTO [Pathdelay-Diagramm] ([Zenith Pathdelay], [Zenith Path], Epoche) "
    SQL = SQL + "SELECT DISTINCTROW ([ " + GN + "].[Epoche]- Int([ " + GN + "].[Epoche]))*24 AS [Zenith Pathdelay], "
    SQL = SQL + "[Pathdelay]*Sin([Elevation]*3.14/180) AS [Zenith Path], "
    SQL = SQL + "Int([ " + GN + "].[Epoche]) As [Epoche]"
    SQL = SQL + "FROM [ " + GN + " ] "
    SQL = SQL + "WHERE ([ " + GN + "].[Pathdelay calc] = Yes) ;"
    Abfrage.SQL = SQL
    Abfrage.Execute
    Abfrage.Close
    DoCmd Hourglass False
End If
DoCmd Echo True, "Datumsliste wird erstellt"
SQL = "SELECT DISTINCT Int([Epoche]) As Datum FROM [ " + GN + " ] "
SQL = SQL + " GROUP BY Int([Epoche]), [ " + GN + "].[Pathdelay calc] HAVING [ " + GN + "].[Pathdelay calc]=Yes ;"
Set Abfragedatum = DB.CreateDynaset(SQL)
DoCmd OpenForm "Pathdelay/Diag nach Tagen", A_Normal, , , A_Edit, A_Normal
Abfragedatum.MoveFirst
Forms![Pathdelay/Diag nach Tagen]![GruppenBezeichnung] = GN
Forms![Pathdelay/Diag nach Tagen]![DatumSelect] = Format(Abfragedatum.Datum, "dd.mm.yyyy")
dddd = DateSerial(Year(Forms![Pathdelay/Diag nach Tagen]![DatumSelect]), Month(Forms![Pathdelay/Diag nach Tagen]![DatumSelect]), Day(Forms![Pathdelay/Diag nach Tagen]![DatumSelect]))
SQL = "SELECT DISTINCTROW [Zenith Pathdelay], [Zenith Path] "
```

```

SQL = SQL + "FROM [Pathdelay-Diagramm] "
SQL = SQL + "WHERE ([Epoche] = " + Str$(ddddd) + ") "
SQL = SQL + "ORDER BY [Zenith Pathdelay] ;"
Set AbfrageSelect = DB.OpenQueryDef("Pathdelay Select")
AbfrageSelect.SQL = SQL
End Function

```

### Prozedur Pathdelay\_Diagramm\_UnterForm wird vom Makro UnterFormPath

```

Function Pathdelay_Diagramm_UnterForm ()

Dim DB As Database
Dim SteuerElement As Control
Dim GN, SQL As String
Dim ddddd As Long

Rem Begin Function
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) *** GruppenName aus Formular übernehmen
dddd = DateSerial(Year(Forms![Pathdelay/Diag nach
Tagen]![DatumSelect]), Month(Forms![Pathdelay/Diag nach
Tagen]![DatumSelect]), Day(Forms![Pathdelay/Diag nach
Tagen]![DatumSelect]))
SQL = "SELECT DISTINCTROW [Zenith Pathdelay], [Zenith Path] "
SQL = SQL + "FROM [Pathdelay-Diagramm] "
SQL = SQL + "WHERE ([Epoche] = " + Str$(dddd) + ") "
SQL = SQL + "ORDER BY [Zenith Pathdelay] ;"
AbfrageSelect.SQL = SQL
DoCmd Requery "Pathdelay-Diagramm"
End Function

```

### Prozedur Pathdelay\_smooth

```

Function Pathdelay_Smooth ()

Dim SteuerTag As Control
Dim SteuerElement As Control
Dim GN, SQL As String
Dim Status, i, j, k As Integer
Dim ddddd As Long
Dim Org As Table
Dim Result As Table
Dim min, max, sum As Double

Rem Begin Function
Set DB = CurrentDB()
Set SteuerElement = Forms![Hauptfenster]![AktData]
GN = SteuerElement.Spalte(1) *** GruppenName aus Formular übernehmen
Set Abfrage = DB.OpenQueryDef("Pathdelay Create")
Rem *** suchen ob Tabellen und aktuelle Datensatzgruppe übereinstimmen
If InStr(60, Abfrage.SQL, GN, 0) = 0 Then
    Status = True
Else
    If MsgBox("Möchten Sie die Datensätze zu den Smoothen aktualisieren?"
        , 292, "Smooth aktualisieren") = 6 Then
        Status = True
    Else
        Status = False
    End If
End If
If Status = True Then
    DoCmd Hourglass True

```

```

DoCmd Echo True, "Alte Daten der Pathdelay-Smoothe löschen"
Abfrage.SQL = "DELETE FROM [Pathdelay-Smooth] ;"
Abfrage.Execute
DoCmd Echo True, "Datentabelle für Pathdelay-Smoothe wird erstellt"
SQL = "Insert INTO [Pathdelay-Smooth] ([Zenith Pathdelay],
    [Zenith Path],Epoche) "
SQL = SQL+"SELECT DISTINCTROW Avg([" + GN + "].[Epoche]-
    Int([" + GN + "].[Epoche]))*24 AS [Zenith Pathdelay], "
SQL = SQL+"Avg([Pathdelay]*Sin([Elevation]*3.14/180))AS [Zenith Path], "
SQL = SQL+"First(Int([" + GN + "].[Epoche])) As [Epoche]"
SQL = SQL+"FROM [" + GN + "]"
SQL = SQL+"WHERE ([ " + GN + "].[Pathdelay calc] = Yes) "
SQL = SQL+"Group by Int([" + GN + "].[Epoche]),
    Int((" + GN + "].[Epoche]-Int([" + GN + "].[Epoche]))*2880) ;"
Abfrage.SQL = SQL
Abfrage.Execute
Abfrage.Close
*** Glätten der Daten ***
Set Org = DB.OpenTable("Pathdelay-Smooth")
Set Result = DB.OpenTable("Pathdelay-Smooth")
For j = 1 To AnzahlIter
    Org.MoveFirst
    Result.MoveFirst
    For i = 1 To (AnzahlPunkte \ 2)
        Result.MoveNext
    Next i
    Do Until Org.EOF
        min = 100: max = 0: sum = 0: k = 0
        For i = 1 To AnzahlPunkte
            If Not Org.EOF Then
                If (Org.[Zenith Path] < MaxWert) And
                    (Org.[Zenith Path] > MinWert) Then
                    If Org.[Zenith Path] < min Then min = Org.[Zenith Path]
                    If Org.[Zenith Path] > max Then max = Org.[Zenith Path]
                    sum = sum + Org.[Zenith Path]
                    k = k + 1
                End If
                Org.MoveNext
            End If
        Next i
        If (k > 2) And (Not Result.EOF) Then
            Result.Edit
            Result.[Zenith Path smooth] = (sum - max - min) / (k - 2)
            Result.Update
        End If
        Result.MoveNext
        If Not Org.EOF Then
            For i = 2 To AnzahlPunkte
                Org.MovePrevious
            Next
        End If
    Loop
Next j
DoCmd Hourglass False
End If
DoCmd Echo True, "Datumsliste wird erstellt"
SQL = "SELECT DISTINCT Int([Epoche]) As Datum FROM [" + GN + "]"
SQL = SQL + " GROUP BY Int([Epoche]), [" + GN + "].[Pathdelay calc]
    HAVING [" + GN + "].[Pathdelay calc]=Yes ;"
Set Abfragedatum = DB.CreateDynaset(SQL)
DoCmd OpenForm "Pathdelay/Smooth nach Tagen",A_Normal, , , A_Edit, A_Normal
Abfragedatum.MoveFirst
Forms![Pathdelay/Smooth nach Tagen]![GruppenBezeichnung] = GN
Forms![Pathdelay/Smooth nach Tagen]![DatumSelect] =

```

```

        Format (Abfragedatum.Datum, "dd.mm.yyyy")
    ddddd = DateSerial (Year (Forms! [Pathdelay/Smooth nach
        Tagen]! [DatumSelect]), Month (Forms! [Pathdelay/Smooth nach
        Tagen]! [DatumSelect]), Day (Forms! [Pathdelay/Smooth nach
        Tagen]! [DatumSelect]))
    SQL = "SELECT DISTINCTROW [Zenith Pathdelay], [Zenith Path],
        [Zenith Path smooth] "
    SQL = SQL + "FROM [Pathdelay-Smooth] "
    SQL = SQL + "WHERE ([Epoche] = " + Str$(dddd) + ") "
    SQL = SQL + "ORDER BY [Zenith Pathdelay] ;"
    Set AbfrageSelect = DB.OpenQueryDef ("Pathdelay Select")
    AbfrageSelect.SQL = SQL
End Function

```

Prozedur Pathdelay\_smooth\_Unterform wird vom Makro UnterFormSmooth aufgerufen

```

Function Pathdelay_smooth_Unterform ()

Dim DB As Database
Dim SteuerElement As Control
Dim GN, SQL As String
Dim ddddd As Long

Rem Begin Function
    Set SteuerElement = Forms! [Hauptfenster]! [AktData]
    GN = SteuerElement.Spalte(1) *** GruppenName aus Formular übernehmen
    ddddd = DateSerial (Year (Forms! [Pathdelay/Smooth nach
        Tagen]! [DatumSelect]), Month (Forms! [Pathdelay/Smooth nach
        Tagen]! [DatumSelect]), Day (Forms! [Pathdelay/Smooth nach
        Tagen]! [DatumSelect]))
    SQL = "SELECT DISTINCTROW [Zenith Pathdelay], [Zenith Path],
        [Zenith Path smooth] "
    SQL = SQL + "FROM [Pathdelay-Smooth] "
    SQL = SQL + "WHERE ([Epoche] = " + Str$(dddd) + ") "
    SQL = SQL + "ORDER BY [Zenith Pathdelay] ;"
    AbfrageSelect.SQL = SQL
    DoCmd Requery "Pathdelay-Smooth"
End Function

```

## Anhang F

### Makro Werkzeug

Um die Prozeduren des Moduls Einlesen ausführen zu können, müssen zuerst die entsprechenden Makros aufgerufen werden.

Makronamen: Werkzeug				
Makrogruppe	Aktion	Bedingung	Argument1	Argument2
neue_Gruppe	AusführenCode		neue_Gruppe()	
	ÖffnenFormular		Hauptfenster	0
	AktualisierenDaten		AktData	

Tabelle 9 : Makrogruppe Werkzeug

Prozedur neue\_Gruppe generiert eine neue Datenbankgruppe (siehe Seite 23).

```
Function neue_Gruppe ()
```

```
Dim DB As Database
Dim Tabelle As Table
Dim GruppenName As String
Dim Listenfeld As Control
```

*Rem begin function*

```
GruppenName = InputBox$("Bitte geben Sie einen neuen Gruppennamen ein.",
    "Gruppennameneingabe")
```

```
If GruppenName <> "" Then
    DoCmd SelectObject A_Table, "Original Tabelle", True
    DoCmd CopyObject , GruppenName
    DoCmd SelectObject A_Table, "Original Tabelle hot load temp", True
    DoCmd CopyObject , GruppenName + " hot load temp"
    DoCmd SelectObject A_Table, "Original Tabelle Tipping to Data", True
    DoCmd CopyObject , GruppenName + " Tipping to Data"
    DoCmd SelectObject A_Table, "Original Tabelle Meteo", True
    DoCmd CopyObject , GruppenName + " Meteo"
    DoCmd SelectObject A_Table, "Original Tabelle Eingabefehler", True
    DoCmd CopyObject , GruppenName + " Eingabefehler"
    DoCmd SelectObject A_Table, "Original Tabelle Tipping LinReg", True
    DoCmd CopyObject , GruppenName + " Tipping LinReg"
    Set DB = CurrentDB()
    Set Tabelle = DB.OpenTable("Verfügbare Daten")
    Tabelle.AddNew
    Tabelle.[DatenName] = GruppenName
    Tabelle.Update
    DB.Close
```

```
End If
```

```
End Function
```