EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VLT PROGRAMME

Accepted by ESO
Technical Archive

ESO Technical
Archive
02.05.2008

VERY LARGE TELESCOPE

OPD Controller (As-built) Software Detailed Design

Doc. No.: VLT-SPE-ESO-15420-4000

Issue: 2

Date: 30/04/2008

Prepared: Bertrand Bauvir / Nicola Di Lieto — Name
30/04/2008 — Date
Signature

Approved: Thanh Phan Duc — Name
30/4/08 — Date
Signature

Released: Francoise Delplancke — Name
30/04/08 — Date
Signature

CHANGE RECORD

ISSUE	DATE	SECTION/PAGE AFFECTED	REASON/INITIATION DOCUMENTS/REMARKS
1/prep.1	14/08/2006	All	First draft
1/prep.2	16/08/2006	All	Second draft prepared for review.
1/prep.3	20/08/2006	4.3	Dedicated configuration database point.
1/prep.4	26/08/2006	4.1	Add instrument OPD offset of the night.
1	31/08/2006	All	First release
2	30/04/2008	All	Update after integration of PRIMA changes and new vibration cancellation algorithm

TABLE OF CONTENTS

1 INTRODUCTION **7**

1.1 Purpose 7

1.2 Scope 7

1.3 Applicable Documents 7

1.4 Reference Documents 8

1.5 Acronyms 8

2 FUNCTIONAL SPECIFICATION **9**

2.1 Overview 9

2.1.1 About OPD Sign Convention 10

2.1.2 Chopping 10

2.2 Software Requirements 10

2.3 Constraints and Performance 11

2.4 Design Decisions 11

3 ARCHITECTURE OVERVIEW **12**

3.1 Hardware 12

3.2 Software 12

3.2.1 Application Package 12

3.2.2 Support Packages 14

3.2.3 CMM Modules 14

4 DESIGN DESCRIPTION **15**

4.1 TAC Real-Time Algorithm 15

4.1.1 TAC Algorithm Structure 15

4.1.2 Fringe Tracking with an Instrument Only (i.e. VINCI or MIDI) 16

4.1.3 Fringe Tracking with FINITO or PRIMA (incl. ZPD Search) 16

4.1.4 Vibration Tracking 18

4.1.5 TAC Specific Function Block 19

4.2 Commands 21

4.2.1 Standard Commands (with application-specific extended behaviour) 21

4.2.2 Public Commands 21

4.2.3 Maintenance Commands 21

4.2.4 Test Commands 21

4.3 Online Database 22

4.3.1 Control Branch 22

4.3.2 Configuration Branch 22

4.3.3 Data Branch 22

5 APPENDIX **24**

5.1 CDT and CIT files 24
5.1.1 opdcPublic.cdt 24
5.1.2 opdcTest.cdt 26
5.1.3 opdcSoftDev.cit 26

5.2 Online Database Description Files 27
5.2.1 opdcDB_CONFIG.class 27
5.2.2 opdcDB_DATA.class 28

TABLE OF FIGURES

Figure 1: Schematic Layout of the Fringe Tracking Facility using FINITO 9
Figure 2: OPDC LCU Hardware structure 12
Figure 3: TAC Algorithm Schematic (1 DL, 1 FTK channel only) 15
Figure 4: FTK state machine 16
Figure 5: ZPD search trajectory 17

This page has been left intentionally blank.

1 Introduction

1.1 Purpose

This document aims to present the design of the OPD Controller Software that is part of the VLTI Fringe Tracking (FTK) Facility.

1.2 Scope

This document shall first shortly describe the VLTI Fringe Tracking Facility and the function of the OPDC LCU therein. It will then present the requirements and constraints associated to this software component. Some design decisions are then issued from these requirements.

In order to ensure compliancy with LCU standard interface and behaviour described in [AD#01], the code of this software component was first generated from LCU Server Framework (LSF) template and extended following the guidelines documented in [AD#03].

The LCC libraries, messaging system and database are documented in [AD#02].

The implementation of the OPDC real-time algorithm relies on Tools for Advanced Control (TAC) application described in [AD#04].

The design description section of this document mainly focuses on the application-specific extensions brought to LSF and on the description of the real-time algorithm.

In addition, the reader should be familiar with VLT software standards.

1.3 Applicable Documents

The following documents, of the issue shown if specified, form part of this manual to the extent specified herein. In the event of conflict between this document and those referenced, the content of this document shall be considered as a superseding requirement unless explicitly stated otherwise herein.

Reference	Document Number	Issue	Date	Title
[AD#01]	VLT-MAN-ESO-17210-0667	1.0	03/12/1996	Guidelines for the Development of VLT Application Software
[AD#02]	VLT-MAN-SBI-17210-0001	3.5	20/10/1999	LCU Common Software User Manual
[AD#03]	VLT-MAN-ESO-17210-2252	2	05/07/2002	LCU Server Framework User Manual
[AD#04]	VLT-MAN-ESO-17210-2970	2	21/03/2004	Tools for Advanced Control User Manual
[AD#05]	VLT-MAN-ESO-17210-3525	1	27/01/2005	RTDScope User Manual
[AD#06]	VLT-SPE-ESO-15400-1954	1.0	09/11/1999	ICD between VLTI Fast Link and DL, OPDC, FSU and Instruments
[AD#07]	VLT-SPE-ESO-15400-4558	1	30/04/2008	Adaptive RLS Vibration Cancellation Algorithm

1.4 Reference Documents

The following documents contain additional information that can be useful to the reader:

Reference	Document Number	Issue	Date	Title
[RD#01]	VLT-SPE-ESO-15430-2790	1	31/08/2006	FINITO Detection Algorithm (As-built) Control Software Detailed Design
[RD#02]	VLT-SPE-ESO-15420-4001	1	31/08/2006	OPD Monitor (As-built) Software Detailed Design

1.5 Acronyms

This document employs several abbreviations and acronyms to refer concisely to an item, after it has been introduced. The following list is aimed to help the reader in recalling the extended meaning of each short expression:

Acronym	Meaning
AD	Applicable Document
ADU	Astronomical Detector Unit
CPU	Central Processing Unit
DD	Design Decision
DL	Delay Line
FTK	Fringe Tracking
LCU	Local Control Unit
LCC	LCU Common Software
LSF	LCU Server Framework
IC	Laboratory Input Channel
IO	Input/Output
ISR	Interrupt Service Routine
OPD	Optical Path Difference
OPDC	OPD Controller
OPDM	OPD Monitor
PPC	Power PC
RD	Reference Document
RMN	Reflective Memory Network
RFM	Reflective Memory Network, 2nd generation
RMS	Root Mean Square
RT	Real-Time
RTC	RT Controller
SNR	Signal to Noise Ratio
SR	Software Requirement
TAC	Tools for Advanced Control
TIM	Time Interface Module
UTC	Universal Time Coordinated
VLT	Very Large Telescope
VLTI	VLT Interferometer
VTK	Vibration Tracking
ZPD	Zero Path Difference

2 Functional Specification

2.1 Overview

Fringe motion happens with earth rotation and any type of disturbance. The latter is not predictable, needs to be measured and compensated to minimize the residual fringe motion seen by the VLTI instruments; it is the purpose of the VLTI Fringe Tracking Facility.

The facility currently encompasses one fringe sensor, i.e. FINITO or PRIMA, a set of Delay Lines (DL) and one dynamic compensator, i.e. the OPD Controller (OPDC). The LCU associated to the previous components are all interconnected by means of a fast communication link featuring real-time capabilities, i.e. the VLTI Fast Link or Reflective Memory Network (RMN). MIDI and VINCI instruments are also present on the RMN to allow some level of fringe tracking without FINITO.

The OPDC does not support concurrent observations; only one fringe sensor is enabled at a time. It is capable of providing FTK for two baselines/three telescopes (i.e. triangle, possibly collapsed) at the same time when using FINITO as a fringe sensor.

The OPDC has been extended to handle one PRIMA FSU, which is handled similarly as one FINITO channel.

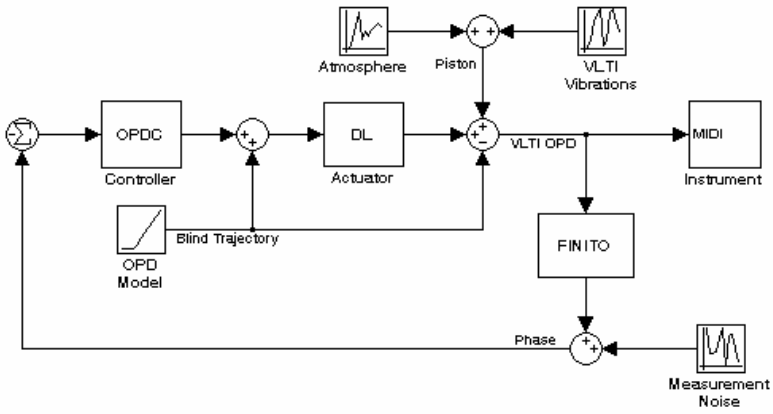

Figure 1: Schematic Layout of the Fringe Tracking Facility using FINITO

The VLTI is suffering from atmospheric piston, but also piston induced by vibrations of components in the optical train; the latter being periodic, it can be modeled and compensated by an adaptive vibration cancellation (a.k.a. tracking, VTK) algorithm (see [AD#07]).

The function of the OPDC is threefold:

- Propagate the VINCI or MIDI instrument fringe offset to the tracking delay line (with adequate sign). In this mode, the OPDC acts only as a switchyard; the instrument drives the tracking DL to search and center fringes.
- Provide FTK capabilities with FINITO or PRIMA. In this mode, the OPDC is responsible to search, detect and track fringes using the signals measured by FINITO/PRIMA.
- Provide VTK capabilities with FINITO/PRIMA.

2.1.1 About OPD Sign Convention

FINITO, PRIMA or the VLTI instruments are providing a fringe position error signal, the sign of which depends on how the beams are routed in the laboratory; i.e. the OPD provided by the instrument/FINITO is defined as the OPL of the beam, propagating in the laboratory, closest to the tunnel minus the OPL of the other beam. E.g. if we consider the case of UT1-DL1-IC1 and UT2-DL2-IC3 feeding MIDI instrument, the OPD computed by MIDI should be OPL(UT1)-OPL(UT2); if the reference DL is DL1, the centering offset is applied to DL2 OPL position with the same sign (positive OPD means the OPL of UT1 beam is greater than the one of UT2, the fringes are centered by increasing DL2 OPL, hence the positive sign); if for operational reasons the reference DL is DL2, the same centering offset should be applied to DL1 with opposite sign.

The OPDC provides a mechanism to change the sign of the fringe centering offset depending on the interferometer configuration.

2.1.2 Chopping

An instrument like MIDI requires chopping to measure the flux of the scientific target. A mode of MIDI considers acquiring fringes with parallel photometry. Therefore, the telescope might be chopping while FTK. The OPDC must suspend the FTK loop when the telescopes are pointing to the sky and resume it when pointing to the target.

The telescopes can operate in two different chopping/guiding mode:

- The telescope points by default to the target and M2 periodically offsets to the sky. The target light is propagated to the VLTI when M2 is in its center position.
- The telescope points by default to the sky and M2 periodically offsets onto the target. The target light is propagated to the VLTI when M2 is in its offset position.

The respect of the chopping cycle is guaranteed provided:

- The start time is an exact UTC second.
- The chopping period and duty cycle yield target/sky time slices that are exact multiples of the OPDC real-time sampling period.

2.2 Software Requirements

The main software requirements (SR) are:

Reference	Requirement
[SR#1]	Propagate instrument fringe searching/centering offset to the interferometer tracking arm.
[SR#2]	Provide VLTI with FTK capability with FINITO/PRIMA sensor.
[SR#3]	Synchronize FTK facility with telescope chopping cycle.
[SR#4]	Publish fringe offset with respect to the blind trajectory by means of DEL.FRINGE.OFFSET ops.log keyword.
[SR#5]	Provide FTK with VTK capability.

2.3 Constraints and Performance

The main constraints and performance requirements associated with this software are:

Reference	Dependency	Decision
[Constr#1]	N/A	Provide a VLT standard interface to the outside world (Command interpreter, Online database Access, Error reporting, etc.).
[Constr#2]	N/A	The process in charge of the fringe tracking algorithm is subject to hard real-time (RT) constraints with high sampling frequency (the design goal being 4kHz). The process shall be activated periodically with a fixed, low jitter, repetition period. No sample loss is allowed.

2.4 Design Decisions

The following list of design decisions (DD) is issued a priori from the above requirements:

Reference	Dependency	Decision
[DD#1]	[Constr#1]	The software will be based on LSF.
[DD#2]	[SR#1] & [Constr#2]	The control algorithm will be implemented using TAC (that features real-time capabilities).
[DD#3]	[Constr#2]	The hardware structure will use 2 Power PC (PPC) boards to reduce the control algorithm activation latency time.
[DD#4]	[Constr#2]	The real-time algorithm shall be triggered at 2kHz by the TIM board interrupt and shall be evaluated at interrupt level to reduce the activation latency time.
[DD#5]	[SR#1] & [SR#2]	The sign of the FTK control signal shall be computed from the current interferometer configuration.
[DD#6]	[DD#5]	Selection of the tracking arm is not the responsibility of the OPDC. In order to compute/apply the adequate OPD sign, the OPDC shall be told which is the DL and IC associated to the interferometer's tracking arm.
[DD#7]	[SR#3]	The telescope chopping cycle is started on an absolute UTC time. The OPDC only needs information on the chopping period, duty cycle and telescope guiding window.

3 Architecture Overview

3.1 Hardware

The following LCU architecture has been selected for this application:

- 2 Motorola PPC CPU boards used in a Master/Slave configuration.
- 1 ESO Time Interface Module (TIM) board.
- 1 VMIC VMIVME-5576 Reflective Memory Network (RMN) board.
- 1 VMIC VMIPMC-5565 Reflective Memory Network (RFM) board. This is a daughter card piggybacked on the Slave PPC CPU board.

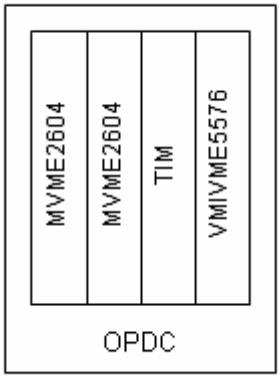

Figure 2: OPDC LCU Hardware structure

3.2 Software

3.2.1 Application Package

This software is responsible for the control of the OPDC. It encompasses the following components:

- A command interface, based on CCS message system and on a command interpreter running on the first CPU board of the LCU.
- A public database providing status and signals information.
- A real-time process running on the second CPU board.
- A graphical engineering user interface.

Compliancy with [AD#01] is ensured by the use of LSF to implement standard behaviour and to generate part of the application-specific code (command handling routines) and database structure. The application software was generated using *lsfConfig* tool, LSF configuration file can be found in the appendix.

Interaction with the real-time algorithm is done through an instance of a LSF software device. The background monitoring is handled by an instance of a LSF task device. The real-time algorithm is exploiting TAC application and its library of function blocks.

3.2.2 Support Packages

The application makes extensive use of the following support packages to implement the standard LCU behaviour and the RT detection algorithm.

Package Description	***Name***
LCU Common Software	lcc
LCU Server Framework	lsf
Tools for Advanced Control	tac
RMN Data Interface	rmac
RFN Data Interface	rmacrfm
Real-time Display	rtdscope

3.2.3 CMM Modules

This application encompasses the following CMM modules:

opdc

This module contains the complete set of files that compose this application software:

- Application-specific software code (command handlers and TAC function blocks).
- Online database classes files.
- Online database configuration files both for the application and TAC branches.
- Command definition and interface tables (CDT and CIT files).
- TAC algorithm configuration file.

VLT-SPE-ESO-15420-4000

This module shall be used to document the historical development of this document. At least all drafts, preparations and releases which are referenced in any other documents shall be archived as an own version within this module.

4 Design Description

This section presents the design of the opdc application software.

4.1 TAC Real-Time Algorithm

4.1.1 TAC Algorithm Structure

The following diagram represents the TAC algorithm implementing the required functions.

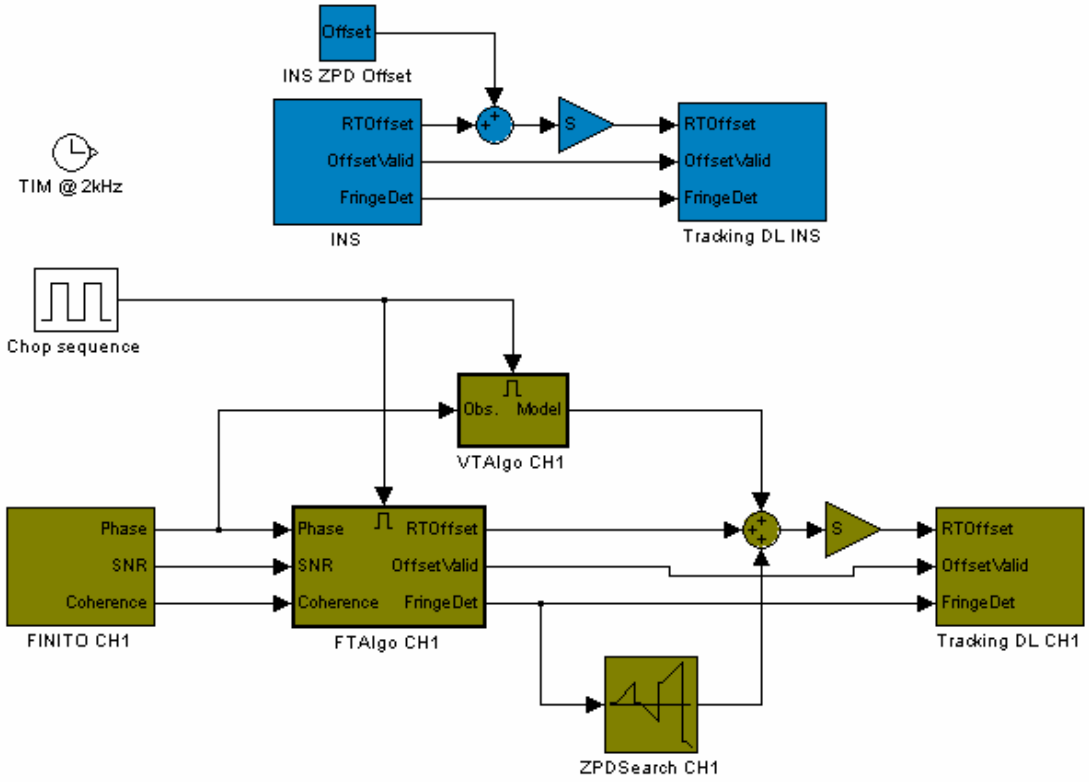

Figure 3: TAC Algorithm Schematic (1 DL, 1 FTK channel only)

The function blocks appearing in green color are instantiated once for fringe sensor channel (i.e. currently FINITO CH1, CH2 and PRIMA FSU).

The algorithm encompasses the following parts:

- An interface to the TIM board to configure periodic interrupts at 2kHz and provide the real-time algorithm with absolute UTC (to enable/disable FTK with the chopping sequence).
- The definition of the telescope chopping sequence. The block is configured with the chopping start (and possibly stop) time, the chopping period, duty cycle and telescope guiding window.
- The instrument interface over RMN to propagate the instrument fringe searching/centering signal to the tracking DL with the adequate sign and a possible offset to cope with static OPD model error.

- The RMN interface blocks to read and write FTK signals.
- The RFN interface blocks to read PRIMA FTK signals.
- The FTK controller (including model based VTK) and Zero Path Difference (ZPD) search.

4.1.2 Fringe Tracking with an Instrument Only (i.e. VINCI or MIDI)

This function encompasses three blocks, the instrument interface over RMN that is either configured to read data from VINCI or MIDI, a gain that is either +1 or -1 depending on the interferometer tracking arm (i.e. +1 if the tracking arm is entering the lab through IC3) and a DL interface over RMN that is connected to the tracking delay line; the parameter of which is the DL identifier (that is set to zero to disable this function).

In addition, a constant block is used to enable static OPD offset compensation (e.g. static OPD model error); this offset is added to the instrument fringe offset signal.

4.1.3 Fringe Tracking with FINITO or PRIMA (incl. ZPD Search)

FTK includes three parts, a state machine driven by the fringe contrast SNR, a central fringe acquisition procedure driven by the coherence and the actual position loop using the fringe phase.

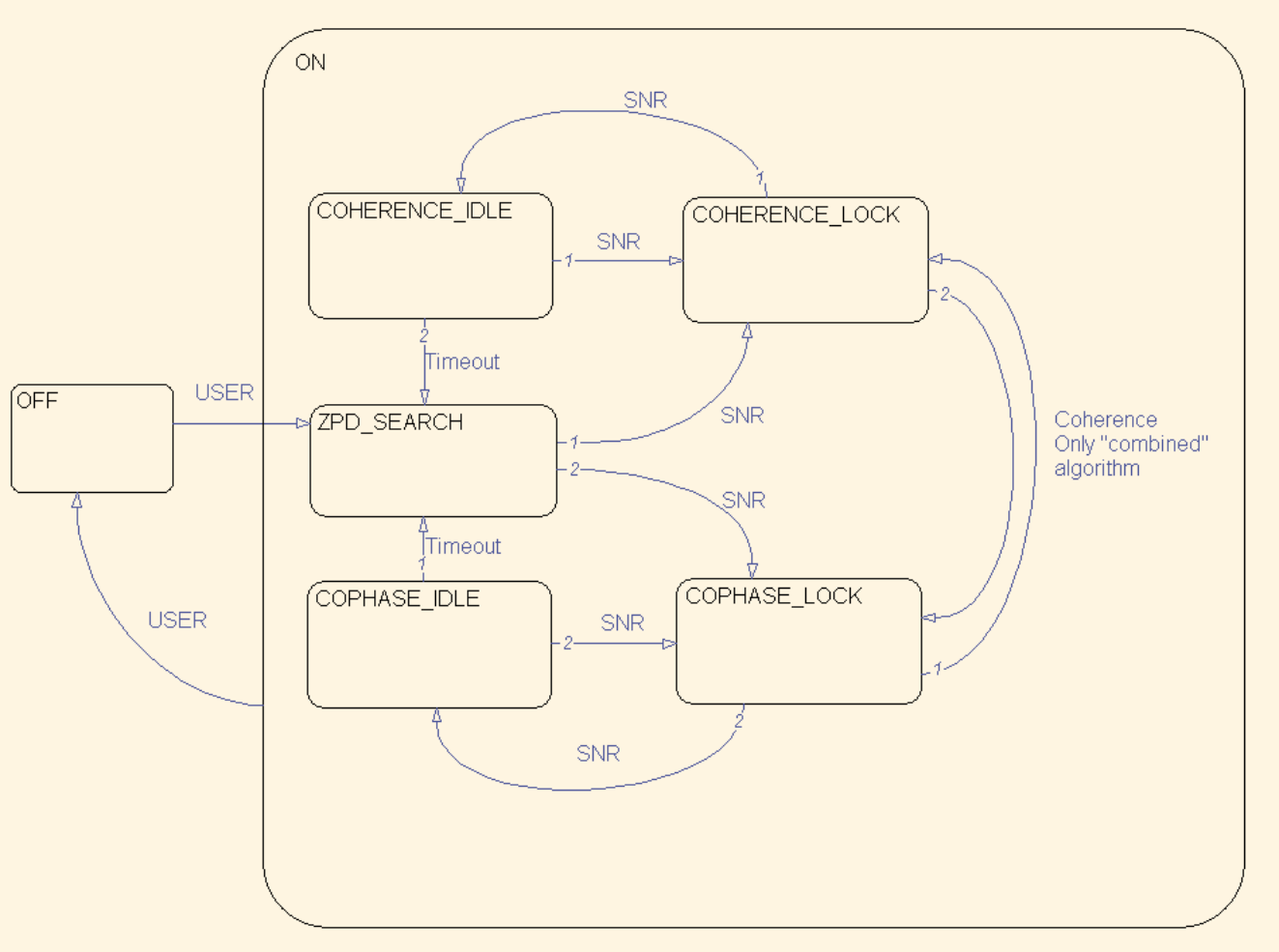

Figure 4: FTK state machine

The FTK state machine transitions happen with FINITO SNR signal. When FTK is started, the OPDC first applies a ZPD search trajectory; i.e. the DL position is offset with respect to the blind tracking trajectory (trajectory compensating the fringe motion due to earth rotation) until the instantaneous SNR goes above a fringe <*detLevel*> threshold. The FTK loop is then closed on the phase signal. The implemented controller is of the form of a discrete transfer function up to the 9th order (it allows to implement any kind of control, e.g. simple PI, or controller design coming from loop shaping or H2/H∞ optimization techniques).

In order to avoid driving the DL on noise, FTK loop is paused/resumed when the filtered (i.e. moving average) SNR signal goes below/above a threshold with hysteresis (i.e. <*openLevel*> < <*closeLevel*> thresholds) or when the chopping state is such that the telescopes are pointing to the sky/target; the states are denoted IDLE and LOCK. If the SNR stays below the <*openLevel*> threshold for a certain <*timeout*>, the loop is stopped and search trajectory is resumed.

Various algorithm types are available:

1. Coherencer algorithm, only using the coherence signal to acquire the central fringe
2. Cophase algorithm, using the phase algorithm (with coherence-based kicks)
3. Combined algorithm, switches between 1) and 2) based on coherence and SNR
4. Group Delay algorithm, which is similar to 2) for use with PRIMA. This algorithm only uses the Cophase states, but instead of tracking Phase=0 it applies a variable offset designed to maintain the PRIMA group delay signal at around zero, which allows tracking of the white fringe.

Ideally, the <*openLevel*> and <*closeLevel*> thresholds are set to ensure the loop opens when the accuracy phase signal becomes low due to noise; the <*timeout*> is favourably linked to the atmosphere coherence time, i.e. the search shall be resumed if the loop has stayed open for a time long enough for the atmosphere to have moved the fringes out of FINITO/PRIMA OPD 'field of view' (i.e. some 15μm).

The initial ZPD search trajectory is a one dimension spiral. When the search is resumed (after fringes lost for some time), the trajectory is triangular with a geometric progression of the amplitude and a constant rate.

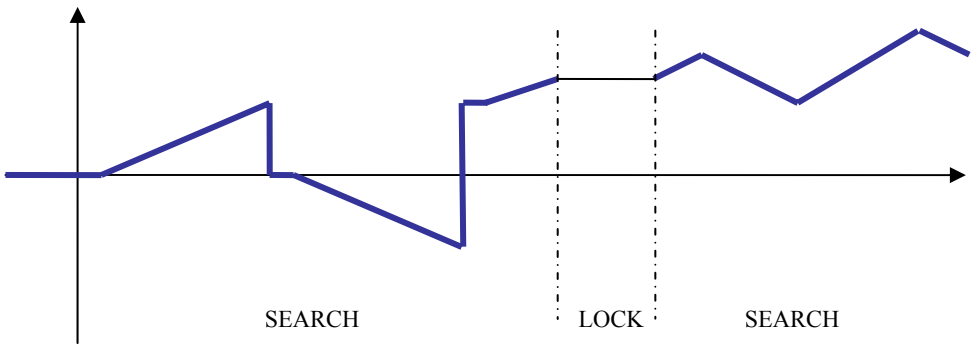

Figure 5: ZPD search trajectory

4.1.4 Vibration Tracking

Due to delay in the loop and the dynamical behaviour of the delay line, high frequency disturbances are not rejected by the FTK loop. Yet, harmonic residual may be observed, modeled as a sinusoidal signal with an amplitude, frequency and phase.

VTK consists of observing the residual phase when the FTK loop is closed and adapting the parameters of the harmonic model in real time. Multiple disturbances may be tracked by multiple instantiation of the function block implementing the adaptive algorithm. Detailed information on the adaptive algorithm is found in [AD#07].

The disturbance model can be used to drive the piezoelectric actuator of the DL to attenuate the harmonic disturbance beyond the rejection bandwidth of the FTK loop.

4.1.5 TAC Specific Function Block

The previous algorithm identifies application-specific one TAC function blocks to be designed in the scope of this software, the definition of the chopping sequence, the FTK and the ZPD search trajectory. The other blocks are part of the standard TAC function block library.

The code of the application-specific TAC function blocks was first generated using *tacCreateBlock* tool.

ChopSynchro block

Type	ChopSynchro
Code file	tacChopSynchroBlock.c
Parameters	<*start*>
	<*stop*>
	<*period*>
	<*dutyCycle*>
	<*guide*>
Inputs	None
Outputs	<*skyOrTarget*>

The block outputs a signal that is equal to one when the telescope is expected to point to the target and zero if the telescope is expected to point to the sky.

FTAlgo block

Type	FTAlgo
Code file	tacFTAlgoBlock.c
Parameters	<*numer*>*1..9*
	<*denom*>*1..9*
	<*detLevel*>
	<*openLevel*>
	<*closeLevel*>
	<*timeout*>
Inputs	<*SNR*>
	<*phase*>
	<*coherence*>
Outputs	<*ftkOffset*>

The specified digital transfer function is applied to the input phase when the FTK loop state machine is in LOCK state, else the output is frozen.

ZPDSearch block

Type	ZPDSearch
Code file	tacZPDSearchBlock.c
Parameters	<*amplitude*>
	<*period*>
	<*offset*>

Inputs <*fringeDet*>
Outputs <*zpdOffset*>

The ZPD search trajectory is applied when the <*fringeDet*> input is zero and frozen else. The <*amplitude*> and <*period*> parameters define the search speed (i.e. typically 0.1mm in 10s).

4.2 Commands

4.2.1 Standard Commands (with application-specific extended behaviour)

Command	Reply	Description
STOP	Ok Error	Suspend FTK and ZPD search.

4.2.2 Public Commands

Command	Parameters	Reply	Description
SETFSEN	<*sensor*>	Ok Error	Identifies which fringe sensor is providing FTK signals. If NONE, all DL control blocks are disabled; MIDI or VINCI will enable the INS part of the algorithm where the OPDC propagates the instrument signals directly to the DL, possibly changing the sign of the DL offset to cope with geometrical characteristics of the interferometer; FNT_CH1, FNT_CH2 or FNT_BOTH will enables one or both FINITO interferometer channels. FSU_A and FSU_B are enabling the PRIMA FSU channel.
SETDLN	<*inputCh*> <*trkDL*> <*sign*>	Ok Error	Identifies the DL and sign for the selected fringe sensor. The <*inputCh*> parameter is used to compute the sign of the OPD offset (the <*sign*> parameter is optional; the <*inputCh*> provides enough information to compute it).
STRTCHP	<*atTime*> <*period*> <*ratio*> <*guide*>	<*time*>	Identifies the start time, cycle characteristics (i.e. period, duty cycle and telescope guiding window) and stop time of the telescope chopping sequence. This information is used to pause/resume FTK when the telescope is pointing to the sky/star.
STOPCHP	<*atTime*>	<*time*>	
STRTFTK	None	Ok Error	Start and stop FTK. The sequence is to start ZPD search trajectory with STRTFTK; the ZPD offset freezes when fringes are detected and FTK loop is closed. When fringes disappear, FTK loop is paused and ZPD search resumes.
STOPFTK	None	Ok Error	

4.2.3 Maintenance Commands

No application-specific maintenance command is identified in the scope of this design.

4.2.4 Test Commands

Command	Parameters	Reply	Description
STRTVTK	None	Ok Error	Start and stop VTK. The frequencies should be listed in a file named opdcVTKFreqList.tbl in the LCU environment directory.
STOPVTK	None	Ok Error	
SETFMOD	<*mode*>	Ok Error	Used to switch between AUTOTEST, AUTOCOLL and SCIENTIFIC modes. The gain of the FTK control loop is set to zero for AUTOTEST, to 0.5 for for AUTOCOLL and 1 for SCIENTIFIC. This function allows testing FTK without having to modify default control parameters.

4.3 Online Database

The online database structure is directly inherited from LSF with the extensions described below.

4.3.1 Control Branch

No application-specific extension to the LSF control branch is forseen in the scope if this application software. Refer to [AD#03] for a description of the control database branch (e.g. the background monitoring task scheduling period).

4.3.2 Configuration Branch

The configuration branch will at least gather the following configuration attributes for each fringe sensor channel (i.e. FNT_CH1 and FNT_CH2):

Point	Attribute	Type	Description
:zpd	.sweep	Double	ZPD search trajectory parameters. The offset is going up to the sweep amplitude value in the specified period and then changes sign.
:zpd	.period	Double	
:zpd	.offset	Double	
:fsm	.detLevel	Double	FTK state machine parameters.
:fsm	.closeLevel	Double	
:fsm	.openLevel	Double	
:fsm	.timeout	Double	
:ctrl	.order	Double	FTK controller (discrete transfer function) order and coefficients.
:ctrl	.numer	Vector	
:ctrl	.denom	Vector	

The database point associated to the instrument configuration contains only a ZPD offset. It allows applying a fixed offset to the blind trajectory of the tracking DL (e.g. hold the OPD offset for the night).

The VTK algorithm is not yet handled automatically by the OPDC.

4.3.3 Data Branch

The data branch will at least gather the following attributes (CH is either INS, CH1, CH2 or PRM):

Point	Attribute	Type	Description
	.fsu	String	Current sensor on RMN.
:CH	.enabled	Boolean	Set if the current channel is selected.
:CH	.tracking	Boolean	FTK running or not.
:CH	.targetDL	Integer	Tracking DL identifier.
:CH	.sign	Double	Sign applied to the DL offset.
:CH	.fringeDet	Boolean	Fringe detected flag.
:CH	.rtOffset	Double	FTK control signal.

:CH	.zpdOffset	Double	ZPD search offset.
:CH	.opdOffset	Double	Total (FTK + ZPD) offset.
:CH	.dlOffset	Double	Applied offset (sign dependant).
:CH:rmn	.snr	Double	Data read on RMN.
:CH:rmn	.phase	Double	
:CH:rmn	.phaseValid	Boolean	
:CH:rmn	.coherence	Double	
:CH:rmn	.coherValid	Boolean	
:chop	.frequency	Double	Relevant chopping parameters.
:chop	.pvRatio	Double	
:chop	.onSky	Boolean	
:chop:start	.seconds	Integer	
:chop:start	.microsec	Integer	
:chop:start	.isoString	String	
:chop:stop	.seconds	Integer	
:chop:stop	.microsec	Integer	
:chop:stop	.isoString	String	

5 Appendix

5.1 CDT and CIT files

5.1.1 opdcPublic.cdt

```
//************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: opdcPublic.cdt,v 2.1 2008/02/13 20:01:12 vltsccm Exp $"
//
// who       when        what
// --------  ----------  ------------------------------------------------
// bbauvir   2005-10-24  created
//------------------------------------------------------------------------

//========================================================================
// Specific
//========================================================================

COMMAND=  SETDLN
SYNONYMS=  opdcSoftDevSetTrackingDelayLine
FORMAT= A
PARAMETERS=
   PAR_NAME= inputCh
   PAR_TYPE= INTEGER
   PAR_RANGE= ENUM 1,3,5,7
   PAR_DEF_VAL= 1

   PAR_NAME= dlId
   PAR_TYPE= INTEGER
   PAR_RANGE= INTERVAL MIN=1;MAX=6
   PAR_DEF_VAL= 1

   PAR_NAME= sign
   PAR_TYPE= INTEGER
   PAR_RANGE= ENUM -1,1
   PAR_DEF_VAL= 1
REPLY_FORMAT= A
HELP_TEXT=
Set tracking DL information.
@

COMMAND=  SETFSEN
SYNONYMS=  opdcSoftDevSetFringeSensor
FORMAT= A
PARAMETERS=
    PAR_NAME= sensor
    PAR_TYPE= STRING
    PAR_RANGE= ENUM "NONE", "VINCI", "MIDI", "FINITO_CH1", "FINITO_CH2", "FINITO_BOTH",
"FSUA", "FSUB", "MET"
    PAR_DEF_VAL= "VINCI"
REPLY_FORMAT= A
HELP_TEXT=
Set name of currently used fringe sensor.
@

COMMAND=  SETFMOD
SYNONYMS=  opdcSoftDevSetFTKMode
FORMAT= A
PARAMETERS=
    PAR_NAME= mode
```

```
    PAR_TYPE= STRING
    PAR_RANGE= ENUM "NONE", "AUTOTEST", "AUTOCOLL", "SCIENTIFIC"
    PAR_DEF_VAL= "AUTOTEST"
REPLY_FORMAT= A
HELP_TEXT=
Set fringe tracking operating mode.
@

COMMAND=  STRTCHP
SYNONYMS=  opdcSoftDevStartChopping
FORMAT= A
PARAMETERS=
    PAR_NAME= atTime
    PAR_TYPE= STRING
    PAR_DEF_VAL= "now"

    PAR_NAME= frequency
    PAR_TYPE= REAL
    PAR_DEF_VAL= 0.0
// Means disable -> period of 0.0 also

    PAR_NAME= pvRatio
    PAR_TYPE= REAL
    PAR_DEF_VAL= 1.0

    PAR_NAME= cycleType
    PAR_TYPE= STRING
    PAR_RANGE= ENUM "A", "B", "BOTH"
    PAR_DEF_VAL= "A"

REPLY_FORMAT = A
REPLY_PARAMETERS=
    PAR_NAME= startTime
    PAR_TYPE= STRING
    PAR_DEF_VAL= "2000-12-31T00:00:00.000000"
HELP_TEXT=
Start Chopping at the specified time.
@

COMMAND=  STOPCHP
SYNONYMS=  StopChopping
FORMAT= A
PARAMETERS=
    PAR_NAME= atTime
    PAR_TYPE= STRING
    PAR_DEF_VAL= "now"

REPLY_FORMAT = A
REPLY_PARAMETERS=
    PAR_NAME= stopTime
    PAR_TYPE= STRING
    PAR_DEF_VAL= "2000-12-31T00:00:00.000000"
HELP_TEXT=
Stop Chopping at the specified time.
@

COMMAND=  STRTFTK
SYNONYMS=  opdcSoftDevStartFringeTracking
FORMAT= A
PARAMETERS=
REPLY_FORMAT = A
REPLY_PARAMETERS=
HELP_TEXT=
Start fringe tracking.
@

COMMAND=  STOPFTK
```

```
SYNONYMS=  opdcSoftDevStopFringeTracking
FORMAT= A
PARAMETERS=
REPLY_FORMAT = A
REPLY_PARAMETERS=
HELP_TEXT=
Stop Fringe tracking.
@
//
//============================================================================
//
// ___oOo___
//
```

5.1.2 opdcTest.cdt

```
//****************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: opdcTest.cdt,v 2.1 2008/02/13 20:01:12 vltsccm Exp $"
//
// who       when        what
// --------  ----------  --------------------------------------------------
// bbauvir   2005-10-24  created
//----------------------------------------------------------------------------

//============================================================================
// Specific
//============================================================================

COMMAND=  STRTVTK
SYNONYMS=  opdcSoftDevStartVibrationTracking
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
REPLY_PARAMETERS=
HELP_TEXT=
Start VTK.
@

COMMAND=  STOPVTK
SYNONYMS=  opdcSoftDevStopVibrationTracking
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
REPLY_PARAMETERS=
HELP_TEXT=
Stop VTK.
@
//
//============================================================================
//
// ___oOo___
//
```

5.1.3 opdcSoftDev.cit

```
//****************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: opdcSoftDev.cit,v 1.10.1.21 2006/06/30 20:40:12 vltsccm Exp $"
//
// who       when        what
// --------  ----------  --------------------------------------------------
// bbauvir   2005-10-24  created
//----------------------------------------------------------------------------
```

```
//
// This file has been generated by a utility
//
//  !!!!!!!!!!  DO NOT MANUALLY EDIT THIS FILE  !!!!!!!!!!
//
//************************************************************************

//========================================================================
// Specific
//========================================================================
SETDLN,  IopdcSoftDevSetDelayLine,           FUNCTION
SETFSEN, IopdcSoftDevSetFringeSensor,        FUNCTION
SETFMOD, IopdcSoftDevSetFTKMode,             FUNCTION
STRTCHP, IopdcSoftDevStartChopping,          TASK, opdSTRTCHP,,,30000,REGISTER
STOPCHP, IopdcSoftDevStopChopping,           FUNCTION
STRTFTK, IopdcSoftDevStartFringeTracking,    FUNCTION
STOPFTK, IopdcSoftDevStopFringeTracking,     FUNCTION
STRTVTK, IopdcSoftDevStartVibrationTracking, FUNCTION
STOPVTK, IopdcSoftDevStopVibrationTracking,  FUNCTION
//
//========================================================================
//
// ___oOo___
//
```

5.2 Online Database Description Files

5.2.1 opdcDB_CONFIG.class

```
// ************************************************************************
// * E.S.O. - VLT project
// *
// * "@(#) $Id: opdcDB_CONFIG.class,v 2.1 2008/02/13 20:01:11 vltsccm Exp $"
// *
// * who       when        what
// * --------  ----------  ----------------------------------------------
// * pduhoux   2003-09-09  created
// *

#include "lsfDB_CONFIG.class"

CLASS "BASE_CLASS" "opdcDB_CFG_RST"
BEGIN
    ATTRIBUTE int32  order 2
    ATTRIBUTE Vector numer(10,double)
    BEGIN
        Value(0,0,0,0,0,0,0,0,0,0)
    END
    ATTRIBUTE Vector denom(10,double)
    BEGIN
        Value(1,0,0,0,0,0,0,0,0,0)
    END
END

CLASS "BASE_CLASS" "opdcDB_CFG_INS_ZPD"
BEGIN
    ATTRIBUTE double offset 0
END

CLASS "opdcDB_CFG_INS_ZPD" "opdcDB_CFG_CH_ZPD"
BEGIN
    ATTRIBUTE double sweep  0
    ATTRIBUTE double period 0
END
```

```
CLASS "BASE_CLASS" "opdcDB_CFG_CH_CTRL"
BEGIN
    ATTRIBUTE int32 type 0
    ATTRIBUTE opdcDB_CFG_RST phase
    ATTRIBUTE opdcDB_CFG_RST coher
END

CLASS "BASE_CLASS" "opdcDB_CFG_CH_FSM"
BEGIN
    ATTRIBUTE double detLevel   0
    ATTRIBUTE double closeLevel 0
    ATTRIBUTE double openLevel  0
    ATTRIBUTE double timeout    0
END

CLASS "BASE_CLASS" "opdcDB_CFG_CH_VTK"
BEGIN
    ATTRIBUTE double dit 0
    ATTRIBUTE bytes64 fileName ""
END

CLASS "BASE_CLASS" "opdcDB_CFG_INS"
BEGIN
    ATTRIBUTE opdcDB_CFG_INS_ZPD zpd
END

CLASS "BASE_CLASS" "opdcDB_CFG_CH"
BEGIN
    ATTRIBUTE opdcDB_CFG_CH_VTK  vtk
    ATTRIBUTE opdcDB_CFG_CH_ZPD  zpd
    ATTRIBUTE opdcDB_CFG_CH_FSM  fsm
    ATTRIBUTE opdcDB_CFG_CH_CTRL ctrl
END

CLASS "lsfDB_CONFIG" "opdcDB_CONFIG"
BEGIN
    ATTRIBUTE opdcDB_CFG_INS INS
    ATTRIBUTE opdcDB_CFG_CH  PRM
    ATTRIBUTE opdcDB_CFG_CH  CH1
    ATTRIBUTE opdcDB_CFG_CH  CH2
END
```

5.2.2 opdcDB_DATA.class

```
// ************************************************************************
// * E.S.O. - VLT project
// *
// * "@(#) $Id: opdcDB_DATA.class,v 2.1 2008/02/13 20:01:11 vltsccm Exp $"
// *
// * who       when        what
// * --------  ----------  ----------------------------------------------
// * pduhoux   2000-04-06  created
// *

#include "lsfDB_DATA.class"

CLASS "BASE_CLASS" "opdcDB_DATA_UTC"
BEGIN
    ATTRIBUTE int32 seconds
    ATTRIBUTE int32 microsec
    ATTRIBUTE bytes32 isoString
END

CLASS "BASE_CLASS" "opdcDB_DATA_RMN_FSU"
```

```
BEGIN
    ATTRIBUTE bytes32 name
    ATTRIBUTE double visibility   0
    ATTRIBUTE double phase        0
    ATTRIBUTE int32  phaseValid   0
    ATTRIBUTE double coherence    0
    ATTRIBUTE int32  cohValid     0
END

CLASS "BASE_CLASS" "opdcDB_DATA_INS"
BEGIN
    ATTRIBUTE bytes32 name "PassThru"
    ATTRIBUTE int32  enabled        0
    ATTRIBUTE double rtOffset       0
    ATTRIBUTE double zpdOffset      0
    ATTRIBUTE double opdOffset      0
    ATTRIBUTE double dlOffset       0
    ATTRIBUTE int32  offsetValid    0
    ATTRIBUTE int32  fringeDetected 0
    ATTRIBUTE int32  targetDL       1
    ATTRIBUTE double sign           1
    ATTRIBUTE opdcDB_DATA_RMN_FSU rmn
END

CLASS "BASE_CLASS" "opdcDB_DATA_CH_DIAG"
BEGIN
    ATTRIBUTE double zpdRatio  0
    ATTRIBUTE double idleRatio 0
    ATTRIBUTE double lockCoherenceRatio 0
    ATTRIBUTE double lockCophaseRatio 0
END

CLASS "BASE_CLASS" "opdcDB_DATA_CH_VTK"
BEGIN
    ATTRIBUTE int32 tracking 0
END

CLASS "opdcDB_DATA_INS" "opdcDB_DATA_CH"
BEGIN
    ATTRIBUTE bytes32 name "Combined"
    ATTRIBUTE int32   tracking 0
    ATTRIBUTE opdcDB_DATA_CH_VTK  vtk
    ATTRIBUTE opdcDB_DATA_CH_DIAG diag
END

CLASS "BASE_CLASS" "opdcDB_DATA_CHOP"
BEGIN
    ATTRIBUTE double frequency 0
    ATTRIBUTE double pvRatio   0
    ATTRIBUTE int32  onSkySeq  0
    ATTRIBUTE int32  onTarget  1
    ATTRIBUTE opdcDB_DATA_UTC start
    ATTRIBUTE opdcDB_DATA_UTC stop
END

CLASS "lsfDB_DATA" "opdcDB_DATA"
BEGIN
    ATTRIBUTE bytes32 fsu  "FINITO_CH2"
    ATTRIBUTE bytes32 mode "SCIENTIFIC"
    ATTRIBUTE opdcDB_DATA_CHOP chopping
    ATTRIBUTE opdcDB_DATA_INS INS
    ATTRIBUTE opdcDB_DATA_CH  PRM
    ATTRIBUTE opdcDB_DATA_CH  CH1
    BEGIN
        ATTRIBUTE opdcDB_DATA_RMN_FSU rmn
        BEGIN
            ATTRIBUTE bytes32 name "FNT_CH1"
```

```
        END
    END
    ATTRIBUTE opdcDB_DATA_CH  CH2
    BEGIN
        ATTRIBUTE opdcDB_DATA_RMN_FSU rmn
        BEGIN
            ATTRIBUTE bytes32 name "FNT_CH2"
        END
    END
    ATTRIBUTE opdcDB_DATA_CH  PRM
    BEGIN
        ATTRIBUTE opdcDB_DATA_RMN_FSU rmn
        BEGIN
            ATTRIBUTE bytes32 name "FSUB"
        END
    END
END
```

___oOo___