

# Lower hybrid current drive control and data acquisition system

Nils Plesner Basse

Plasma Science and Fusion Center Massachusetts Institute of Technology USA
--



12TH OF FEBRUARY 2003

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Overview of data to be supplied</b>	<b>5</b>
2.1	Low power circuit amplitude and phase control . . . . .	5
2.2	Calibration tables . . . . .	6
<b>3</b>	<b>Overview of data to be acquired</b>	<b>7</b>
3.1	Active controller system . . . . .	7
3.2	Transmitter protection system . . . . .	7
3.3	Coupler protection system . . . . .	8
3.3.1	CPS cards . . . . .	8
<b>4</b>	<b>Hardware overview</b>	<b>10</b>
4.1	CompactPCI . . . . .	10
4.2	Computers . . . . .	11
<b>5</b>	<b>The lower hybrid MDSplus tree</b>	<b>12</b>
5.1	Existing structure . . . . .	12
5.2	Proposed structure . . . . .	19
5.2.1	First level . . . . .	19
5.2.2	Second level . . . . .	19
5.2.3	Third level . . . . .	19
5.3	Create, write and read shot . . . . .	19
<b>6</b>	<b>LHCOSY: Structure and inner workings</b>	<b>21</b>
6.1	Data management using MDSplus . . . . .	21
6.1.1	cPCI card data acquisition . . . . .	22
6.1.2	cPCI card triggering . . . . .	22
6.1.3	cPCI card testing . . . . .	24
6.2	The IDL routines . . . . .	24
6.2.1	LH_CONTROL.PRO . . . . .	24

<i>CONTENTS</i>	3
6.2.2 RF_CONTROL.PRO . . . . .	24
<b>7 LHCOSY: How to operate</b>	<b>25</b>
<b>8 Physics issues</b>	<b>26</b>
8.1 High bootstrap fraction plasmas . . . . .	26
8.2 Edge density profiles . . . . .	26
8.3 Suitable target plasmas . . . . .	26

# Chapter 1

## Introduction

A lower hybrid current drive (LHCD) system is currently being installed on the Alcator C-Mod tokamak. Initially, the system is being designed to operate at 4.6 GHz and deliver 3 MW of power in a pulse lasting up to 5 seconds. 12 klystrons yielding 250 kW each are to be used, situated on 3 carts in the Alcator C-Mod cell (i.e., the torus hall).

The main thrust of my work is to construct the lower hybrid control system (LHCOSY) software, and to acquire and manage the resulting data.

These notes are intended to be a collection of useful technical information and basic physics considerations with regard to the LHCD operation.

# Chapter 2

## Overview of data to be supplied

The operator has to choose the value of several parameters. Those parameters should have externally imposed safety limits set elsewhere, prohibiting operation that could potentially damage the hardware components. Additionally, a limited set of operators should be allowed.

### 2.1 Low power circuit amplitude and phase control

- Pulse start and stop time with respect to the CAMAC t0 (?) trigger
- Amplitude waveforms: Control the total injected power
- Phase waveforms: Control the spectrum
- Boolean value: If 'false', the ACS should not use the data. Set to 'false' initially, 'true' after all parameters have been written to the tree and 'false' some time thereafter.
- (Input amplitude and phase waveforms for each klystron. These are to be converted to I and Q values, thereafter to be transferred to the ACS.)

An in-phase and quadrature (I and Q) signal from a complex signal  $K$  having an amplitude  $A$  and a phase  $\Phi$ :

$$K_j(t) = A_j(t) \times e^{i\Phi_j(t)}, \quad (2.1)$$

where  $j$  is the klystron number (1 to 12) and  $t$  is time, is the real and imaginary part:

$$\begin{aligned} \operatorname{Re}(K_j(t)) &= I_j(t) = A_j(t) \times \cos \Phi_j(t) \\ \operatorname{Im}(K_j(t)) &= Q_j(t) = A_j(t) \times \sin \Phi_j(t) \end{aligned} \quad (2.2)$$

This means that the following relations exist:

$$\begin{aligned} A_j(t) &= \sqrt{I_j^2(t) + Q_j^2(t)} \\ \Phi_j(t) &= \arctan \left( \frac{Q_j(t)}{I_j(t)} \right) \end{aligned} \quad (2.3)$$

For simplest operation, the phase and amplitude will not be time dependent. Does it matter what the absolute phase is at the outer columns?

## 2.2 Calibration tables

Detectors associated with the TPS and CPS systems supply a voltage. Ideally, each detector should have a calibration table, so that the power can be found by knowing the voltage.

Calibration of each detector could be done by writing software sweeping the signal generator input power and measuring the resulting voltage.

As an initial step, some detector characteristics could be measured manually. An average could be constructed and used for all detectors of the same type.

Initial tests were made using a HP83732B synthesized function generator.

# Chapter 3

## Overview of data to be acquired

### 3.1 Active controller system

Based on information in [4]. Initially 2 active controller racks will be used, a 3rd to be made when the 4th cart is added.

- Get ready
- Gate
- Start
- 9 kHz gated clock
- $8 + 4 = 12$  I and Q analogue inputs from the I and Q detectors
- $8 + 4 = 12$  I and Q analogue outputs to the vector modulators

This amounts to a total of 28 signals from each rack, or 56 signals in total. Note that the 'Gate' and 'Start' signals will collapse into one signal, reducing the number of signals to 54.

2 additional signals, a 'Data acquisition trigger' and a 'Data acquisition clock' will be acquired. All in all this brings us back to a total of 56 signals.

### 3.2 Transmitter protection system

Signals from the TPS will be arriving from the 3 carts to racks below the carts.

- 4 forward power detectors

- 4 reverse power detectors
- 4 pulse current (?)
- 4 body current

So 16 signals from each rack. In addition the high voltage current and voltage will be measured at one cart. This means a total of 50 signals.

### 3.3 Coupler protection system

12 klystrons in total (3 carts). Power detectors for each klystron:

- 1 forward, 1 reverse power detector at the 'Intermediate directional coupler' between the klystron and the 'Power divider 1A'
- 4 forward, 4 reverse power detectors after the 'Power divider 1B' and the 'Power divider 1C'
- 4 forward, 4 reverse power detectors at the individual waveguides

This totals 18 power detectors per klystron, so 216 detectors in total.

Questions:

1. Does this agree with what is stated in [1]?
2. Why does it say 'LOAD' instead of 'FWD' in Monty's drawing?
3. What does Monty mean by '...a total of 60 forward and 156 reverse power sample points' in [2]?

#### 3.3.1 CPS cards

A study has previously been made concerning the settings on the CPS cards (see files VSWR1.xls and VSWR.xls). A number of definitions are necessary here:

- Voltage standing wave ratio (VSWR): In a transmission line (e.g. a waveguide), the ratio of maximum to minimum voltage in a standing wave pattern. I.e., a superposition of two counter propagating waves. A  $VSWR = \frac{V_{max}}{V_{min}} = \frac{V_{fwd} + V_{rev}}{V_{fwd} - V_{rev}} = 1$  means that the voltage transfer has a maximum, i.e. perfect impedance match (VSWR is always greater than or equal to 1).



- $P[\text{dBm}] = 10 \log_{10} \left( \frac{P[\text{W}]}{0.001} \right)$  is the conversion between power in dBm and W.
- The reflection coefficient  $\rho = \frac{\text{VSWR}-1}{\text{VSWR}+1}$ .
- The return loss =  $10 \log_{10} \left( \frac{P_{\text{fwd}}}{P_{\text{rev}}} \right) = -20 \log_{10}(\rho)$ .

In the actual system, the forward power detection is attenuated by  $-50-30 = -80$  dB and the reverse power detection is attenuated by  $-50-24 = -74$  dB.

Assumptions:

1. Forward power in W
2. VSWR
3. Forward and reverse voltages are measured
4. Forward and reverse gains

The goal is to select the VSWR and forward and reverse gains, so that the final voltage difference is close to constant for a large variation of the forward power.

# Chapter 4

## Hardware overview

### 4.1 CompactPCI

A total of 21 cPCI cards and 8 chassis were bought, see table 4.1. The cards are model ACQ32CPCI(?) made by D-TACQ. A card has 32 channels, each can be sampled up to 250 kHz. 16 bit resolution meaning up to 0.5 MB of data per channel per second. A total of 128 MB of memory, so the minimum time window is 8 seconds. They have the following front connections:

- Channel 1-16 ( $18 + 19 = 37$  pins)
- Channel 17-32 ( $18 + 19 = 37$  pins)
- Trig out (Lemo)
- Trig in (Lemo)
- Clk out (Lemo)
- Clk in (Lemo)

'Clk in' and 'Clk out' are soldered together, so input can be supplied to either one.

The critical issue at the moment is whether we have a sufficient number of channels for the CPS. We have 96 channels (3 cards), but the number of signals originally meant to be acquired is 216. If we need more channels, S.Wukitch is supposed to buy those.

Additional cPCI cards are installed for ethernet connection. They are model CPV5350 manufactured by Motorola.

Used by	Cards	Chassis
-	Total of 13	Total of 6
Wukitch	5 (ours)	1 (ours)
ACS	2	1
TPS	3	3
CPS	3	1

Table 4.1: Whereabouts of cPCI cards and chassis.

## 4.2 Computers

The full LH acquisition computer names (see table 4.2) are to be postfixed by '.psfc.mit.edu' if referred to from outside MIT.

Location	Computer name	IP address	Card number(s)
Setup lab	lh1	198.125.177.234	-
Mech lab	lh_control	198.125.178.66	-
ACS	pcdaqh1	198.125.176.133	21003, 21006
CPS	pcdaqh2	198.125.176.66	21004, 21005, 21012
PXI1	-	169.254.0.26	-
PXI2	-	169.254.0.27	-
TPS(3)	pcdaqh3	198.125.176.69	21014
TPS(2)	pcdaqh4	198.125.176.70	21015
TPS(1)	pcdaqh5	198.125.176.71	21013

Table 4.2: Acquisition computers.

The cPCI status of each computer can be monitored by typing the computer name in the 'Address' field of a web browser.

# Chapter 5

## The lower hybrid MDSplus tree

The lower hybrid tree ('LH') can be opened using the following command: **traverser -tree lh -shot -1**. The '-1' command means that the model file is opened. For a description of traverser, see [www.mdsplus.org](http://www.mdsplus.org). Traverser can be used to add or remove parts of a tree.

Currently, the following persons have write access to the LH tree:

N.P.Basse, J.Bosco, N.Kambouchev, R.Parker, S.Sherman, J.A.Stillerman, D.Terry.

### 5.1 Existing structure

The structure as created by N.Kambouchev and J.A.Stillerman in 2002:

- Devices: DT\_5, DT\_8, J221, LH\_DECODER, LH\_ENCODER  
(actually added by J.A.Stillerman and N.P.Basse)
- Numeric: TSTART
- Children: ACTION, CONSTANTS, VARIABLES

Figure 5.1 shows the overall tree structure; the ANALYSIS child has been added by N.P.Basse. The structures are stated above; the TSTART numeric contains the value -4 seconds.

Figure 5.2 shows the ACTION child. Data is passed to and from Labview through this child. The child has three TEXT nodes. The ACK node contains the time and date of the latest acknowledgement. The START node contains one of several possible strings; each string is read by Labview and a corresponding command is executed (start, run, pause, retrieve):

1. Start: Write I and Q data to the PXI chassis.



Figure 5.1: Overview of the LH tree.

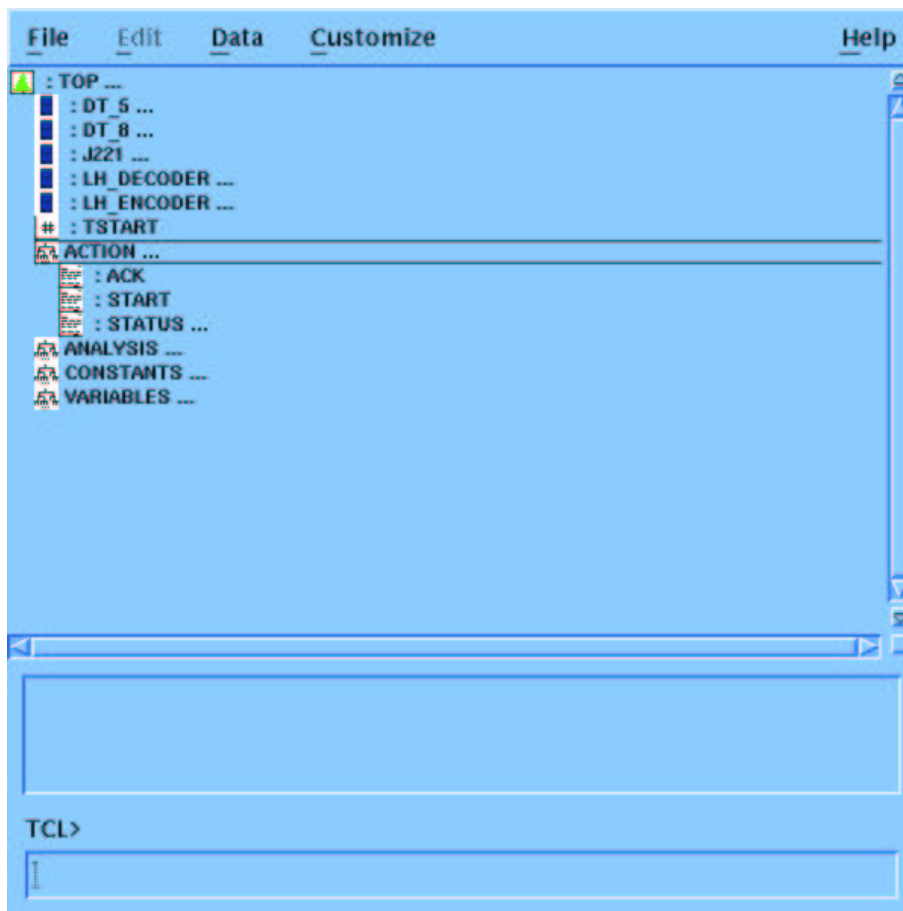


Figure 5.2: ACTION child.

2. Run: Make the shot.
3. Pause: Enable reading of the data.
4. Retrieve: Read the data.

The STATUS node contains a string displaying whether the previous command was executed successfully. STATUS has an additional sub-node called ERR\_DESC, containing a description of an error made during the course of the program.

Figure 5.3 shows the CONSTANTS child. The child has 3 numeric nodes. The EN\_DIS (enable/disable) node contains an array of 12 values, each having the value 0 or 1. 0 means that the klystron is off, 1 that it is on. The MAP (mapping) node is currently not used; intended use is for mapping of cables. The O\_C (open/closed) node contains an array of 12



Figure 5.3: CONSTANTS child.

values, each having the value 0 or 1. 0 means that the system is running in open loop, 1 that it is closed loop. The sub-child PIDGAINS contains 3 numeric nodes D (derivative), I (integral) and P (proportional). Each of those contains an array of 24 values.

Figure 5.4 shows the VARIABLES child. It contains 5 numeric nodes and 12 sub-children. We begin by describing the 5 nodes. The IPS node is not currently used, but intended to contain IP addresses of the various computers comprising the system. The LADDSQUARE node contains an array of 24 values; the values determine the additional size of boxes in the I-Q plane that are allowed values by the control system. The LENGTH node is not implemented? The LSQUARESIZE node contains an array of 24 values; the values determine the initial box size in the I-Q plane. The node ZEROOFFSET contains an offset for the 12 klystrons?



Figure 5.4: Overview of the VARIABLES child.



The first of the 12 sub-children of the VARIABLES child is shown in figure 5.5. Since the sub-children have a lot of entries, we simply list them:

- Numeric AGAIN: 1
- Signal AMPLITUDE: `resample(A_SET_POINTS, 0., 5., 1/9000.)`
- Numeric AOFFSET: 0
- Signal A\_SET\_POINTS: `Build_Signal([datapoints],*,[timepoints])`
- Signal I: `Build_Signal(IGAIN * 10-((AMPLITUDE * AGAIN + AOFFSET)/20) * COS(PHASE * PGAIN + POFFSET) + IOFFSET,*, TIMEBASE)`
- Numeric IGAIN: 5.
- Signal INCOMINGI: `Build_Signal(Set_Range(datapoints),*,TIMEBASE)`
- Signal INCOMINGQ: `Build_Signal(Set_Range(datapoints),*,TIMEBASE)`
- Numeric IOFFSET: 0
- Signal OUTGOINGI: `Build_Signal(Set_Range(datapoints),*,TIMEBASE)`
- Signal OUTGOINGQ: `Build_Signal(Set_Range(datapoints),*,TIMEBASE)`
- Numeric PGAIN: 1.
- Signal PHASE: `resample(P_SET_POINTS, 0., 5., 1/9000.)`
- Numeric POFFSET: 0
- Signal P\_SET\_POINTS: `Build_Signal([datapoints],*,[timepoints])`
- Signal Q: `Build_Signal(QGAIN * 10-((AMPLITUDE * AGAIN + AOFFSET)/20) * SIN(PHASE * PGAIN + POFFSET) + QOFFSET,*,TIMEBASE)`
- Numeric QGAIN: 1.
- Numeric QOFFSET: 0.
- Axis TIMEBASE: Start = 0, End = 5., Increment = 1/9000.



Figure 5.5: VARIABLES child for the first klystron (K00). The other 11 entries (K01...K11) are identical in structure.

## 5.2 Proposed structure

### 5.2.1 First level

HARDWARE, ANALYSIS, RESULTS

### 5.2.2 Second level

The suggestion from J.A.Stillerman is to make branches for each physical location. The cPCI cards will be at 3 separate locations:

- ACS: In the setup lab
- TPS: In the cell
- CPS: In the cell

Even though the TPS and CPS systems are both in the cell, they will be physically separated.

HARDWARE branches: ACS, TPS, CPS

ANALYSIS branches: INPUT, CALIBATION, structures to analyse raw data(?), INTERMEDIATE RESULTS

Results branches: FORWARD POWER, REVERSE POWER, PLASMA COUPLING, FINAL AMPLITUDES, FINAL PHASES

### 5.2.3 Third level

ACS branches: 2 cPCI cards. These cards are called 'DT\_1' and 'DT\_2' at the moment. The 'DT' refers to the fact that the cards were built by D-TACQ. The numbering convention has now been changed: The cards are now numbered after which slot they are placed in. Slot numbers run from 1 to 8, from right to left.

TPS branches: 3 cPCI cards

CPS branches: 3(?) cPCI cards

## 5.3 Create, write and read shot

Several small IDL programs have been written to access the LH tree.

- **WRITE** create\_lh\_shot.pro: This creates a shot in the LH tree. If the shot existed before the call, the tree retains the data stored previously.

- **WRITE** fire\_lh\_shot.pro: This program writes data to the LH model tree (calls write\_lh\_en\_dis.pro).
- **READ** knots\_resampled.pro: Reads and plots amplitude knots and resampled amplitudes from K00 for the LH model tree.
- **READ** plot\_lh\_shot.pro: This program reads and plots data acquired using the cPCI cards for a given shot.
- **READ** read\_lh\_iq.pro: This program reads and plots incoming and outgoing I and Q values for a given shot.
- **READ** read\_lh\_shot.pro: This program reads data from the LH tree for a given shot. Optionally (an IF loop), the amplitudes and phases are plotted and other values printed.
- **READ** test\_labview.pro: This program opens a widget allowing the user to write or monitor values in the LH tree for a given shot.
- **READ/WRITE** write\_lh\_en\_dis.pro: Reads amplitudes and the EN\_DIS array from the LH model tree. If all amplitudes are zero, the corresponding EN\_DIS value is set to zero. Otherwise it is set to one.
- **WRITE** write\_lh\_shot.pro: This program writes default waveforms to the LH model tree. It differs from the fire\_lh\_shot.pro in that the waveforms are specified in the program instead of being passed on from the header (calls write\_lh\_en\_dis.pro).

Maybe a check should be included in fire\_lh\_shot.pro and write\_lh\_shot.pro to check whether all amplitudes and phases before TSTART and after TSTOP are equal to zero.

## Chapter 6

# LHCOSY: Structure and inner workings

The idea is that the lower hybrid control system (LHCOSY) is a Linux based application, running IDL with a widget user interface. A restricted set of users is allowed to access the program.

For a list of simple IDL widget programs, type 'wexmaster' on the commandline.

Inputs from the operator would be those described in chapter 2. The calibration tables would remain until a new calibration is performed.

The challenge is how to handle the acquired data described in chapter 3.

The main purposes would be:

- If one or more klystrons are shut down during the pulse due to signals from the TPS or CPS (or both), plots have to be shown that are able to display what went wrong.
- The coupling quality to the plasma can be assessed by analysing the forward and reverse power acquired by the TPS and CPS systems.
- All of the inter shot analysis has to be done on a timescale of 5 minutes in order to be able to make necessary adjustments before the next discharge.

### 6.1 Data management using MDSplus

Traverser

Acquisition, storage, how to delete, how to access, how to trigger, clock, how to connect signal generator

### 6.1.1 cPCI card data acquisition

How to initialise a given (test) shot number:

1. `mdstcl` (opens a tree command line TCL>)
2. `set tree lh /shot=-1` (opens the model tree)
3. `create pulse <shot number>`
4. `set tree lh /shot=<shot number>`
5. `do /method dt_<slot> init` (initialise cPCI card in a given slot)

After the initialisation has been made, a hardware trigger is connected to the cPCI board. For testing purposes, a trigger box is used. Eventually, the boards will be triggered from the CAMAC timing system.

How to store a given (test) shot number:

1. `do /method dt_<slot> store` (assuming that you have a TCL prompt)

After these operations (see `/home/basse/idl/lhcosy/init.pro` and `/home/basse/idl/lhcosy/store.pro`), the data acquired is stored in the tree and can be accessed using standard MDSplus commands. An example can be found here: `/home/basse/idl/lhcosy/plot_lh.pro`.

### 6.1.2 cPCI card triggering

How to configure the cPCI cards for triggering using 'traverser'.

#### One card

Situation here: One card is externally triggered.

1. `traverser -tree lh -shot -1&`
2. Highlight a card, right click and choose 'Setup Device'
3. Choose 'Trigger Routing'
4. In the 'DI2' row, set 'Wire' to 'MI2 ai trig-F' and 'Bus' to 'PXLTRIG2'. The 'Time' box sets the label for the first sample.

## Two cards

Situation here: First card is externally triggered, second card is triggered through the mezzanine at the back of the board. Follow the steps for one card for the first card. For the second card, do the following:

1. Highlight the second card, right click and choose 'Setup Device'
2. Choose 'Trigger Routing'
3. In the 'DI2' row, set 'Wire' to 'None' and 'Bus' to 'PXL-TRIG2'. The 'Time' box is set to 'DT-<slot of first card>:DI2'.

## CAMAC triggering

Additional hardware was added to the LH tree:

- J221: The Jorway Model 221 12 channel timing and sequence module. (There is also a Jorway Model 222 TTL buffer module in the crate.) The output 12 (actually from the J222) is connected to 'Trig In' on the DT.8 cPCI card.
- LH\_DECODER: The output E0 is connected to 'Clk In' on the DT.8 cPCI card.
- LH\_ENCODER: The 'Tx' is connected to the decoder 'Rx'.
- TSTART:

Some modifications were made to the init.pro program and an additional triggering program trig.pro was made. Note that at the moment the init.pro, trig.pro and store.pro sequence only works on the VMS system. (It is only the trig.pro program that does not work on Linux.) If a problem using these programs occurs where the error includes 'no logical name match', try the following at the VMS command line: **ccl remcam**.

How to configure the new hardware using Traverser.

J221: Right-click, choose 'Setup Device'. The 'Trigger' and 'External Clock' fields are specified. The output channel 12 is selected; left-click on the 'Modify' button to see the expression written.

LH\_DECODER: Right-click, choose 'Setup Device'.

LH\_ENCODER: Right-click, choose 'Setup Device'.

TSTART: Right-click, choose 'Display Data'.

### 6.1.3 cPCI card testing

Test acquisition using a function/signal generator to input e.g. sine waves at low voltage levels.

The function generator is connected to the DT.8 cPCI card, pin 17. 'Trig out' from the DT.8 cPCI card is connected to pin 18.

## 6.2 The IDL routines

### 6.2.1 LH\_CONTROL.PRO

In N.Kambouchev's VMS directory (the root), a program called `lh_control.pro` is to be found. Running this opens a window where 12 amplitudes and 12 phases are drawn. These can be modified and saved to the LH tree.

### 6.2.2 RF\_CONTROL.PRO

Y.Lin made a simplified version of `rf_control.pro` for me to work on: `rf_for_nils.pro`. It runs on VMS. To run it after logging on to the VMS do the following:

1. `set default user10:[basse.idl.rf]`
2. `idl`
3. `.r rf_for_nils`
4. `rf_control`

Check

<http://www.psfc.mit.edu/cmod/unix/UNIXhelp1.3/Pages/VMStoUNIX.html>  
for a Linux to VMS command conversion table.



## Chapter 7

# LHCOSY: How to operate

# Chapter 8

## Physics issues

### 8.1 High bootstrap fraction plasmas

### 8.2 Edge density profiles

Brian LaBombard, probes. Edge density critical for LH launch.

### 8.3 Suitable target plasmas

# Bibliography

- [1] Design of a compact lower hybrid coupler for Alcator C-Mod, S.Bernabei et al., ? (2002)
- [2] The Alcator C-Mod lower hybrid current drive experiment transmitter and power system, M.Grimes et al., IEEE (2002)
- [3] CompactPCI based data acquisition with MDSplus, J.A.Stillerman et al., Fusion Eng. Des. **60** (2002) 241
- [4] The Alcator C-Mod lower hybrid current drive experiment low power microwave and active control system, D.Terry et al., IEEE (2002)