Nanoscan M-Squared Automation

Yudong Sun

Jun 03, 2021

CONTENTS:

1 Indices and tables 5

Python Module Index 7

Index 9

`fit_functions.omega_z(`*params*, *z*`)`
Beam Radii Function to be fitted, according to https://docs.scipy.org/doc/scipy/reference/odr.html

Parameters

params [array_like] rank-1 array of length 4 where `beta = array([w_0, z_0, M_sq, lmbda])`

z [array_like] rank-1 array of positions along an axis

Returns

y [array_like] Rank-1, calculated beam-radii of a single axis based on given parameters

`class fitter.Fitter(`*x*, *y*, *xerror*, *yerror*, *func=<function omega_z>*`)`
The Fitter class fits the given data using scipy.odr

Parameters

x [array_like] Rank-1, Independent variable

y [array_like] Rank-1, Dependent variable, should be of the same shape as `x`

xerror [array_like or function] Rank 1, Error in x, should be of the same shape as `x` or func(x) –> xerror

yerror [array_like or function] Rank 1, Error in y, should be of the same shape as `y` or func(y) –> yerror

func [function, optional] fcn(beta, x) –> y, by default *self.omega_z* (Guassian Beam Profile function)

Methods

fit(initial_params)	Fit the data using the odr Model and saves the output to `self.output`
load_data(x, y, xerror, yerror)	Load the data into a data object
printOutput()	Prints the output of .fit(), otherwise raises a warning

`fit(`*initial_params*`)`
Fit the data using the odr Model and saves the output to `self.output`

Parameters

initial_params [array_like] Represents the initial guesses. Rank 1 Array with length equal to the number of parameters defined for self.model.For w(z): Rank 1 of length 4 with `initial_params = array([w_0, z_0, M_sq, lmbda])`

Returns

self.output [Output instance] This object is also assigned to the attribute .output of Fitter

`load_data(`*x*, *y*, *xerror*, *yerror*`)`
Load the data into a data object

Parameters

x [array_like] Rank 1, Independent variable

y [array_like] Rank 1, Dependent variable, should be of the same shape as `x`

xerror [array_like or function] Rank 1, Error in x, should be of the same shape as `x` or func(x) –> xerror

yerror [array_like or function] Rank 1, Error in y, should be of the same shape as y or func(y) –> yerror

`printOutput()`
Prints the output of .fit(), otherwise raises a warning

Raises

RuntimeWarning Raised if .fit() has not been run.

`class stage.controller.Controller(devMode=True)`
Abstract Base Class for a controller

Methods

KeyboardInterruptHandler(signal, frame)	Abort and close the serial port if interrupted.
startSignalHandlers()	Starts appropriate signal handlers to handle e.g.

abort	
closeDevice	

`KeyboardInterruptHandler(signal, frame)`
Abort and close the serial port if interrupted. Handles a SIGINT according to https://docs.python.org/3/library/signal.html#signal.signal.

Parameters

signal [int] signal number

frame [signal Frame object] Frame objects represent execution frames. They may occur in traceback objects (see below), and are also passed to registered trace functions.

`startSignalHandlers()`
Starts appropriate signal handlers to handle e.g. keyboard interrupts. Ensures safe exit and disconnecting of controller.

`class stage.controller.GSC01(*args, **kwargs)`
Class for the GSC-01 Controller Microcontroller Model: OptoSigma GSC-01

Currently the device is to CENTRAL HOME, i.e. the origin is the center of the stage.

Methods

KeyboardInterruptHandler(signal, frame)	Abort and close the serial port if interrupted.
closeDevice()	Closes the serial device connection
initializeDevice()	Initializes the serial devices and saves it into self.dev
loadConfig([devConfig])	Load the config for device communication from either a json file or a dictionary into self.cfg
send(cmd[, waitClear, raw, waitTime])	Sends a command to the GSC-01 Controller
startSignalHandlers()	Starts appropriate signal handlers to handle e.g.

abort	
read	
waitClear	

`closeDevice()`
Closes the serial device connection

`send(`*cmd, waitClear=False, raw=False, waitTime=0*`)`
Sends a command to the GSC-01 Controller

Parameters

cmd [Union[bytearray, str]] If `raw = True` then cmd is a `bytearray` that is directly sent to the controller. Otherwise, cmd is a string command that is encoded into ASCII before being sent to the controller.

waitClear [bool, optional] [description], by default False

raw [bool, optional] Flag for whether the input command is a bytearray or string, by default False

waitTime [float, optional] Waiting time in seconds before writing to the device, by default 0. Can be used to cool down.

Returns

output [Union[bytearray,int]] Returns 0 if `self.devMode = True` else returns the results from `self.read()`

`class stage.controller.SerialController(`*devConfig=None, *args, **kwargs*`)`
Abstract Base Class for a serial controller

Methods

`KeyboardInterruptHandler`(signal, frame)	Abort and close the serial port if interrupted.
`closeDevice`()	Closes the serial device connection
`initializeDevice`()	Initializes the serial devices and saves it into self.dev
`loadConfig`([devConfig])	Load the config for device communication from either a json file or a dictionary into self.cfg
`startSignalHandlers`()	Starts appropriate signal handlers to handle e.g.

abort	
read	
send	

`closeDevice()`
Closes the serial device connection

`initializeDevice()`
Initializes the serial devices and saves it into self.dev

Raises

RuntimeError Raised if unable to establish serial communication

`loadConfig(`*devConfig=None*`)`
Load the config for device communication from either a json file or a dictionary into self.cfg

Parameters

devConfig [Union[dict,str,None], optional] json file or dictionary of configuation details, by default None

Raises

RuntimeError Raised if an invalid config file is found but self.devMode = False

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

f

fit_functions, 1
fitter, 1

s

stage.controller, 2

INDEX

C

closeDevice() (*stage.controller.GSC01 method*), 2
closeDevice() (*stage.controller.SerialController method*), 3
Controller (*class in stage.controller*), 2

F

fit() (*fitter.Fitter method*), 1
fit_functions
    module, 1
fitter
    module, 1
Fitter (*class in fitter*), 1

G

GSC01 (*class in stage.controller*), 2

I

initializeDevice() (*stage.controller.SerialController method*), 3

K

KeyboardInterruptHandler() (*stage.controller.Controller method*), 2

L

load_data() (*fitter.Fitter method*), 1
loadConfig() (*stage.controller.SerialController method*), 3

M

module
    fit_functions, 1
    fitter, 1
    stage.controller, 2

O

omega_z() (*in module fit_functions*), 1

P

printOutput() (*fitter.Fitter method*), 2

S

send() (*stage.controller.GSC01 method*), 3
SerialController (*class in stage.controller*), 3
stage.controller
    module, 2
startSignalHandlers() (*stage.controller.Controller method*), 2