Package ‘LMR’

October 13, 2023

Type Package

Title Theoretical modelling of lossy mode resonance (LMR) based fiber optic temperature sensor

Version 1.0.0

Author Krzysztof Gajowniczek, Phd, DSc

Maintainer Krzysztof Gajowniczek <krzysztof_gajowniczek@sggw.edu.pl>

Description Lossy mode resonance (LMR) is a physical phenomenon recently exploited for fiber optic sensing. LMR-based devices are widely used for detecting refractive index variation, humidity, pH (acidity or basicity of an aqueous solution), chemical or biological species, gases or even voltage. Two main types of geometries can be distinguished: prism-based and waveguide-based. In both cases, however, the manufacture as well as prototype development of the sensor requires a very precise and at the same time expensive technology. Therefore, reliable and fast modeling of these devices is desired to reduce costs of their investigation, designing and production. LMR is an R-toolbox for simulating LMR sensors of straight-core geometry (which is the most common type of waveguide-based one). The mathematical model is based mainly on geometrical optics. In addition, for determining the reflection coefficients from stratified media, the equations of classical electromagnetism and transfer matrix method were used.

License GPL-3 + file LICENSE

URL https://github.com/KrzyGajow/LMR

BugReports https://github.com/KrzyGajow/LMR/issues

Depends R (>= 3.5.0)

Imports DT,
doRNG,
doSNOW,
GA,
foreach,
Hmisc,
magrittr,
plotly,
purrr,
pracma,
readr,

shiny,
shinyjs

Encoding UTF-8

Language en-US

RoxygenNote 7.1.2

Suggests knitr,
rmarkdown,
parallel,
doParallel,
testthat (>= 3.0.0)

VignetteBuilder knitr

LazyData true

Config/testthat/edition 3

R topics documented:

`consoleLMR` *consoleLMR*

Description

Theoretical modelling of lossy mode resonance (LMR)

Usage

```
consoleLMR(waveMin, waveMax, waveStep, coreD, L, dL, Layers, angleMax)
```

Arguments

`waveMin`	Wavelength start [um].
`waveMax`	Wavelength end [um].
`waveStep`	Wavelength step [um].
`coreD`	Core diameter [um].
`L`	Length of the modified region [um].
`dL`	Thickness of the coating layers [nm].
`Layers`	Layers.
`angleMax`	Maximum skewness angle [deg].

Details

consoleLMR

Value

Data frame with the wavelength and real part of the transmittance.

Examples

```
## Not run:
waveMin <- 0.85
waveMax <- 1.65
waveStep <- 0.01
temperature <- 300
coreD <- 200
L <- 40000
angleMax <- 90
thickLayers <- c( 220 )

Cladding <- function( lambda, temperature ){

 a1 <- 0.6961663
 a2 <- 0.4079426
 a3 <- 0.8974794
 b1 <- 0.0684043
 b2 <- 0.1162414
 b3 <- 9.896161
 cladding <- sqrt(1.0+(a1*lambda^2/(lambda^2-b1^2))+(a2*lambda^2/(lambda^2-b2^2))+(a3*lambda^2/(lambda^2-b3^2))

 thermal_drift <- 1+1.28e-5 * temperature
 thermal_drift <- 1
 cladding <- cladding * thermal_drift

 return( cladding )

}
Core <- function( lambda, temperature, cladding ){

 core <- 1.0036 * cladding

 return( core )

}
L1 <- function( lambda, temperature ){

 lambda <- lambda*10^(-6)
 epsinf <- 3.5
 tau <- 6.58e-15
 omegap <- 1.533e15
 vlight <- 3*10^8
 omega <- 2.0 * pi * vlight/ lambda
 eps <- epsinf - omegap^2 / ( omega^2 + 1i*(omega/tau) )
```

```
 res <- sqrt(eps)

 thermal_drift <- 1.0 - 1.49e-4 * temperature
 thermal_drift <- 1
 res <- res * thermal_drift

 return( res )

}

Layers <- data.frame( lambda = seq( waveMin, waveMax, by = waveStep ) )
Layers$Cladding <- Cladding( Layers$lambda, temperature )
Layers$Sensing <- 1
Layers$Core <- Core( Layers$lambda, temperature, Layers$Cladding )
Layers$Layers1 <- L1( Layers$lambda, temperature )

Layers$Sensing <- 1.436
Result436 <- consoleLMR( waveMin, waveMax, waveStep, coreD, L, thickLayers, Layers, angleMax )
Layers$Sensing <- 1.422
Result422 <- consoleLMR( waveMin, waveMax, waveStep, coreD, L, thickLayers, Layers, angleMax )
Layers$Sensing <- 1.321
Result321 <- consoleLMR( waveMin, waveMax, waveStep, coreD, L, thickLayers, Layers, angleMax )

plot( x = Result436$Wavelength, y = Result436$Transmittance, ylim = range( c(Result436$Transmittance,
                                                                     Result422$Transmittance,
                                                                  Result321$Transmittance) ),
     type = "l", xlab = "Wavelength", ylab = "Transmittance", col = "blue" )
lines( x = Result422$Wavelength, y = Result422$Transmittance, col = "green" )
lines( x = Result321$Wavelength, y = Result321$Transmittance, col = "purple" )

## End(Not run)
```

RefInd *List of refractive indices*

Description

List of refractive indices

Usage

```
RefInd
```

Format

A data frame with 67 rows and 4 columns:

Name

lambda

Index

Info

Source

<https://en.wikipedia.org/wiki/List_of_refractive_indices>

`runShinyLMR` *runShinyLMR*

Description

Function to run Shiny application

Usage

```
runShinyLMR(launch.browser = TRUE)
```

Arguments

`launch.browser`	By default Shiny application is opened in the default browser.

Details

runShinyLMR

Examples

```
## Not run:
runShinyLMR()

## End(Not run)
```

Index