3 SENSOR DESIGN

The key specification is the sensitivity of the sensor. As biomarkers in breath has their concentrations ranged in ppm to ppb, a highly sensitive sensor must be invented. And sensitivity itself, is the constraint of designing a sensor.

In order to solve the problem, optical fibre sensors are being explored for VOC sensing application, especially the singlemode-multimode-singlemode (SMS) optical fibre structure. The critical factor of using optical fibre is to let light propagating inside the optical fibre interacts with outside medium, which contains VOCs. By using no-core fibre for multimode segment of the structure, the propagating light is allowed to interact with surrounding environment. Hence, multimode segment is served as a sensing region for the structure. On top of that, the structure is relatively easy fabrication method, and it is selected as a method of interest in this research [10]. Surface functionalisation also enhance sensitivity of the sensor [10-12].

3.1 Overview

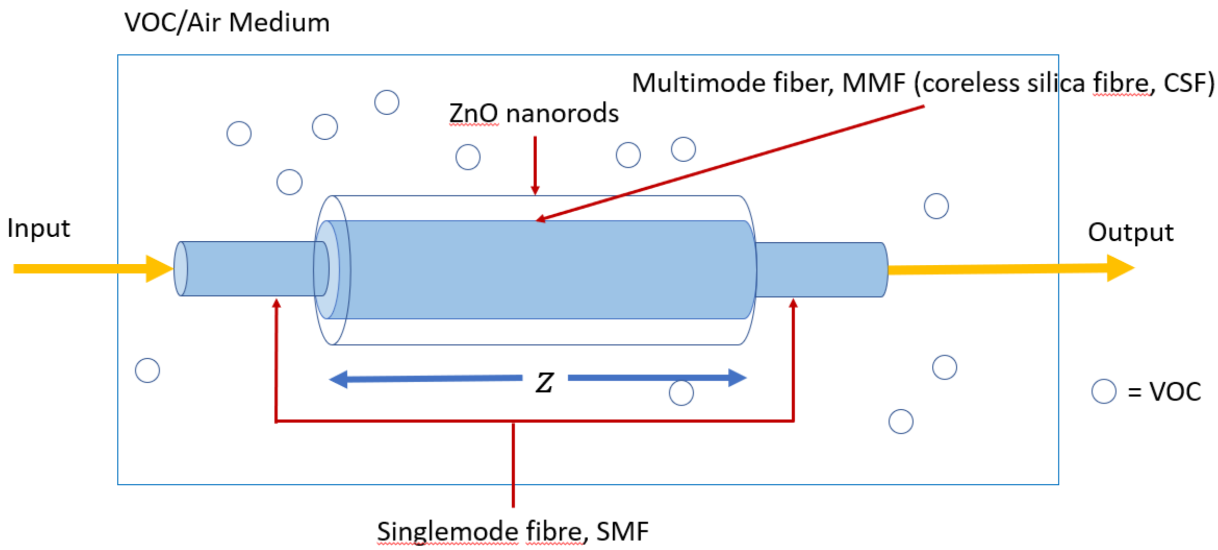

Figure 3.1 Design of the sensor. Composed of ZnO-nanorod-coated on the CSF inserted in between two singlemode fibres.

The design is based on singlemode-multimode-singlemode (SMS) fibre structure as shown in Figure 3.1. It is consisted of 2 types of optical fibres, which are multimode and singlemode fibres. Singlemode fibres serve as a waveguide to deliver the light from light source to the sensor and then to the spectrometer, which records intensity spectra. The multimode fibre serves as a sensing region of the sensor. Coreless silica fibre (CSF) is multimode fibre component, which allows more light interactions at the surface of the sensor, due to the higher number of propagation modes inside the fibre, enhancing evanescent field at the interface [14,15]. As the medium has changes in concentration of VOC, light can sense the change and resulting in change of intensity spectra. This region is also coated with ZnO, as it is reported to improve sensitivity of optical sensor [11,12].

4 EXPERIMENTAL SETUP AND METHODS

4.1 Numerical investigation and simulation

The singlemode-multimode-singlemode (SMS) fibre structure is illustrated in Figure 3.1. It is composed of a multimode fibre (MMF) segment inserted in between two segments of singlemode (SMF) fibres. The theories necessary for the simulations, propagation mode calculation and multimode interference theory is covered, as well as calculation of medium's refractive index calculation.

4.1.1 Propagation mode calculation

In optical fibre, electric field at a specific position inside the fibre can be calculated from the equation (1) [14] assuming on-axis coupling,

$$R(r) = \begin{cases} AJ_l\left(\frac{ur}{a}\right) \; for\ r \leq a, \\ CK_l\left(\frac{wr}{a}\right) \; for\ r \geq a, \end{cases} \tag{1}$$

where r is radial position of an optical fibre. a is a radius of propagating region. J_l and K_l are lth order of Bessel function of the first kind and lth order modified Bessel function of the second kind, while A and C are fitting constants. Due to circular symmetry of light propagating through the singlemode fibre, it can be assumed that the modes in the SMS fibre structure is radially symmetrical, and that lth order Bessel functions are assumed to be 0th order only. Therefore, (1) can be rewritten as

$$R(r) = \begin{cases} AJ_0\left(\frac{ur}{a}\right) \; for\ r \leq a, \\ CK_0\left(\frac{wr}{a}\right) \; for\ r \geq a. \end{cases} \tag{2}$$

where u and w can be defined as

$$u^2 = k_0^2 a^2 \left(n_{core}^2 - n_{eff}^2\right), \tag{3}$$

$$w^2 = k_0^2 a^2 \left(n_{eff}^2 - n_{cladding}^2\right). \tag{4}$$

Here, k_0 is a wavenumber of a light in a vacuum. n_{core} and $n_{cladding}$ are refractive indexes of core and cladding of the fibre, respectively. For the coreless silica fibre (CSF), n_{core} is the refractive index of the glass fibre region, and $n_{cladding}$ is substituted with effective index of the surrounding medium. n_{eff} is an effective index of the whole optical fibre system.

Since the electric field must be continuous for all of domain r, we obtain 2 boundary conditions from (2) which are,

$$R(a-0) = R(a+0), \tag{5}$$

$$\left.\frac{\partial R(r)}{\partial r}\right|_{r=a-0} = \left.\frac{\partial R(r)}{\partial r}\right|_{r=a+0}. \tag{6}$$

and thus,

$$AJ_0(u) - CK_0(w) = 0, \tag{7}$$

$$AuJ_0'(u) - CwK_0'(w) = 0. \tag{8}$$

Through (7) and (8), the effective indexes can be calculated. After substitution of effective indexes into (2), intensity of electric fields propagating through an optical fibre is obtained.

4.1.2 Multimode interference and reimaging distance

When the light is coupled from SMF into MMF, the multiple-mode light propagation is occurred in the MMF. These excited multimodes of propagation can interfere with each other, which is defined as multimode interference. The interference governs the intensity at the output of the SMS structure, which is defined by coupling efficiency, η [15]. The coupling efficiency determines the loss of the intensity at the output compared with the input and can be calculated from

$$\eta = \sum_{j=0}^{M-1}\sum_{h=0}^{M-1} \tilde{a}_j^2 \tilde{a}_h^{*2} exp\left[i\left(\beta_j - \beta_h\right)z\right], \tag{9}$$

where

$$\tilde{a}_j = a_j \sqrt{\frac{P_{j,h}}{P_s}}, \tag{10}$$

$$a_{j,h} = \frac{\int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} E_s(r,\theta) \times R_{j,h}(r,\theta)^* \, rdrd\theta}{P_{j,h}}, \tag{11}$$

$$P_s = \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} |E_s(r,\theta)|^2 rdrd\theta, \tag{12}$$

$$P_{j,h} = \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} \left|R_{j,h}(r,\theta)\right|^2 rdrd\theta, \tag{13}$$

$$\beta_{j,h} = k_0 n_{eff}. \tag{14}$$

Denoted as j or h is a parameter of j^{th} or h^{th} propagation mode of the output. $\tilde{a}_j$ and a_j represent modified expansion coefficient and expansion coefficient of the j^{th} mode of propagation, respectively. The two parameters show how the power of the field expanded between input field (E_s) and output field (R_j) when there are two optical fibres connected. Representing wave number of the light in propagating z-direction, β is called propagating constant and is a product of the wave number in vacuum, k_0, and effective index, n_{eff}, of the j^{th} mode.

The equation (9) provides a relationship between coupling efficiency, η, and propagation length, z. The propagation length that yields the maximum coupling efficiency is defined as reimaging distance, which is used as the length of the no-core fibre for further calculation of the sensor.

5 RESULTS AND DISCUSSION

5.1 Simulation results

The result of the relationship between propagation distance (length of the sensor) and coupling efficiency, using the wavelength of 633 nm and volume fraction of ZnO of 0.05 is shown in Figure 5.1. The graph is similar to a reported research [15]. Reimaging distance is calculated to be 14.28 cm. However, at this length, it is not practical for fabricating the sensor and hence, the length of 5.5, 10.2, and 10.3 cm with acceptable coupling efficiency of -12.56, -8.38, -9.88 dB are selected instead.

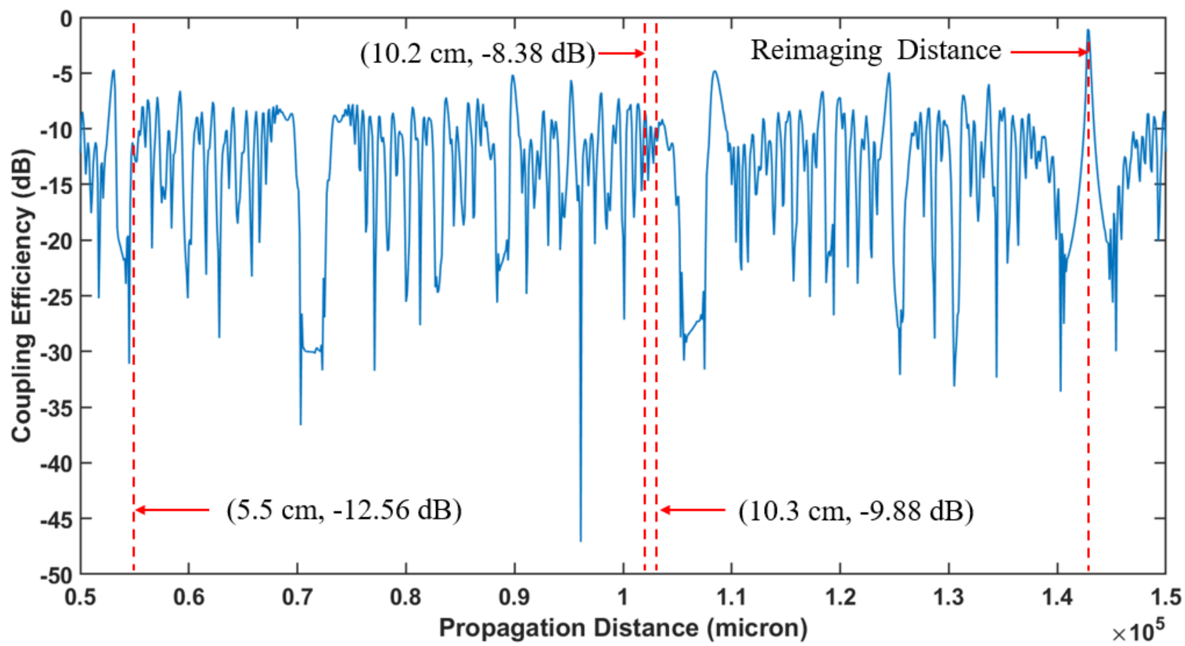

Figure 5.1 Graph of coupling efficiency (dB) and propagation distance (micron) with the length of 5.5 cm, 10.2 cm, 10.3 cm, and reimaging distance indicated.

APPENDIX

I. Matlab code for numerical investigation

i. Main

```
clear all
close all
clc
%---------------------------------------------------------------
--------------%
global ncoresmf ncladdingsmf asmf ncoremmf ncladdingmmf ammf
zmmf epscoresmf epscladdingsmf epscoremmf epscladdingmmf k0
neffarrsmf lambda nmedium aair
ncoresmf = 1.4504; %core refractive index of smf
ncladdingsmf = 1.447; %cladding refractive index of smf
asmf = 4.1; %radius of smf
ncoremmf = 1.443; %refractive index of no core mmf
nmedium = 1.04641231840654; %medium refractive index
ncladdingmmf = nmedium; %refractive index of medium surrounding
no core mmf
ammf = 62.5; %radius of no core mmf
zmmf = 57713; %length of mmf segment
epscoresmf = ncoresmf^2; %core pzermittivity of smf
epscladdingsmf = ncladdingsmf^2; %cladding permittivity of smf
epscoremmf = ncoremmf^2; %permittivity of no core mmf
epscladdingmmf = ncladdingmmf^2; %permittivity of medium
surrounding no core mmf
% lambda = 1.55; %wavelength
lambda = 0.633;
k0 = 2*pi/lambda; %wave number
aair = 87.5; %radius of medium
%----------------------finding mode in smf----------------------
----------%
increment = 0.0001;
ifinal = ((ncoresmf - ncladdingsmf)/increment);
neffarrsmf = zeros(1,round(ifinal)); %empty array of effective
index of smf to be assigned in the next loop
for i = 0:ifinal
     try
        chk = fzero(@characeqnsmf,[(ncladdingsmf +
(i*increment)),(ncladdingsmf + ((i+1)*increment))]);
     catch
        neffarrsmf(i+1) = 0;
        continue
     end
        neffarrsmf(i+1) = fzero(@characeqnsmf,[(ncladdingsmf +
(i*increment)),(ncladdingsmf + ((i+1)*increment))]);
end
```

```
neffarrsmf = neffarrsmf(neffarrsmf ~= 0);
neffarrsmf = neffarrsmf(length(neffarrsmf));

rarrsmf = -asmf:0.01:asmf; %array of r from -asmf to asmf
usmf = findusmf(neffarrsmf); %u of smf
wsmf = findwsmf(neffarrsmf); %w of smf
rarrcladding1 = -ammf:0.01:(-asmf-0.01); %array of r from -ammf
to -asmf
rarrcladding2 = (asmf + 0.01):0.01:ammf; %array of r from asmf
to ammf
rarrair1 = -aair:0.01:(-ammf-0.01); %array of r from -aair to -
ammf
rarrair2 = (ammf + 0.01):0.01:aair; %array of r frin ammf to
aair
rarrall = [rarrair1 rarrcladding1 rarrsmf rarrcladding2
rarrair2]; %array of r from -aair to aair

ecoresmf = besselj(0,(usmf.*rarrsmf./asmf)); %field where -asmf
< r < asmf
c = (besselj(0,usmf))/(besselk(0,wsmf)); %scaling coefficient
ecladdingsmf2 = c.*(besselk(0,(wsmf.*rarrcladding2./asmf)));
%field where asmf < r < ammf
ecladdingsmf1 = zeros(size(ecladdingsmf2)); %field where -ammf <
r < -asmf
for nn = 1:length(ecladdingsmf1)
    ecladdingsmf1(nn) = ecladdingsmf2(length(ecladdingsmf2) -
(nn - 1));
end
eair1 = zeros(size(rarrair1)); %field where -aair < r < -ammf
eair2 = zeros(size(rarrair2)); %field where ammf < r < aair
eair1(:) = ecladdingsmf1(1);
eair2(:) = ecladdingsmf2(length(ecladdingsmf2));
eall = [eair1 ecladdingsmf1 ecoresmf ecladdingsmf2 eair2];
%field where -aair < r < aair
iall = zeros(size(eall)); %field intensity where -aair < r <
aair
for nn = 1:length(eall)
    iall(nn) = (abs(eall(nn)))^2;
end
plot(rarrall,eall)
%----------------------finding mode in mmf----------------------
----------%
ifinal = ((ncoremmf - ncladdingmmf)/increment);
neffarrmmf = zeros(1,round(ifinal)); %empty array of effective
indices of mmf to be assigned in the next loop
for ii = 0:ifinal
```

```
    try
        chk = fzero(@characeqnmmf,[(ncladdingmmf +
(ii*increment)),(ncladdingmmf + ((ii+1)*increment))]);
    catch
         neffarrmmf(ii+1) = 0;
         continue
    end
         neffarrmmf(ii+1) = fzero(@characeqnmmf,[(ncladdingmmf +
(ii*increment)),(ncladdingmmf + ((ii+1)*increment))]);
end
neffarrmmf = neffarrmmf(neffarrmmf ~= 0);
for p = 1:(length(neffarrmmf))/2
    temp = neffarrmmf(p);
    neffarrmmf(p) = neffarrmmf(length(neffarrmmf) - (p - 1));
    neffarrmmf(length(neffarrmmf) - (p - 1)) = temp;
end
plot(neffarrmmf)
%------------------------finding u and w for each mode in mmf---
-------------------------------%
uarrmmf = zeros(size(neffarrmmf)); %u of each mode in mmf
warrmmf = zeros(size(neffarrmmf)); %w of each mode in mmf
carr = zeros(size(neffarrmmf)); %scaling coefficient
for j = 1:length(neffarrmmf)
    uarrmmf(j) = findummf(neffarrmmf(j));
    warrmmf(j) = findwmmf(neffarrmmf(j));
    carr(j) = (besselj(0,uarrmmf(j)))/(besselk(0,warrmmf(j),1));
end
%------------------------finding field and its intensity for
each mode in mmf-----------------------------------%
earrallmmf = zeros(length(neffarrmmf),length(rarrall)); %field
of each mode in mmf

for k = 1:length(neffarrmmf)
    for m = 1:length(rarrall)
        if (rarrall(m) > ammf)
            earrallmmf(k,m) =
carr(k)*(besselk(0,(warrmmf(k)*rarrall(m)/ammf),1));
        elseif (rarrall(m) < -ammf)
            earrallmmf(k,m) =
carr(k)*(besselk(0,(warrmmf(k)*rarrall(length(rarrall) - (m -
1))/ammf),1));
        else
            earrallmmf(k,m) =
besselj(0,(uarrmmf(k)*rarrall(m)/ammf));
        end
    end
end
```

```
iarrallmmf = zeros(size(earrallmmf)); %intensity of each mode in
mmf
for nn = 1:length(neffarrmmf)
    for mm = 1:length(rarrall)
        iarrallmmf(nn,mm) = (abs(earrallmmf(nn,mm)))^2;
    end
end
%-----------------------finding variable pj and aj for each mode
in mmf-----------------------------------------%
pj = zeros(size(neffarrmmf)); %empty array of parameter pj
aj = zeros(size(neffarrmmf)); %empty array of parameter aj
for nn = 1:length(neffarrmmf)
    funpj1 = @(r)
((abs(besselj(0,(uarrmmf(nn).*r/ammf)))).^2).*r; %function of pj
where r is between 0 and ammf
    funpj2 = @(r)
((abs(carr(nn).*besselk(0,(warrmmf(nn).*r/ammf)))).^2).*r;
%function of pj where r is greater than ammf
    pj(nn) = ((integral(funpj1,0,ammf)) +
(integral(funpj2,ammf,Inf))).*2.*pi;
end

integ1 = zeros(size(neffarrmmf)); %empty arrays for storing
integrals in the next loop
integ2 = zeros(size(neffarrmmf));
integ3 = zeros(size(neffarrmmf));
for nn = 1:length(neffarrmmf)
    funaj1 = @(r)
((besselj(0,(usmf.*r./asmf))).*(conj(besselj(0,(uarrmmf(nn).*r/a
mmf))))).*r; %function where r is between 0 and asmf
    funaj2 = @(r)
((c.*besselk(0,(wsmf.*r./asmf))).*(conj(besselj(0,(uarrmmf(nn).*
r/ammf))))).*r; %function where r is between asmf and ammf
    funaj3 = @(r)
((eair2(length(eair2))).*(conj(carr(nn).*besselk(0,(warrmmf(nn).
*r/ammf))))).*r; %function where r is greater than ammf
    integ1(nn) = (integral(funaj1,0,asmf)); %integrate from 0 to
asmf
    integ2(nn) = (integral(funaj2,asmf,ammf)); %integrate from
asmf to ammf
    integ3(nn) = (integral(funaj3,ammf,Inf)); %integrate from
ammf to inf
    aj(nn) = (integ1(nn) + integ2(nn) +
integ3(nn)).*2.*pi./pj(nn);
end
```

```
%-----------------------finding efield of mmf with interference-
--------------------------------------%
betammf = neffarrmmf.*k0; %array of propagation constants of
each mode
% zarr = 57000:10:60000; %array of propagation length
zarr = 50000:100:150000;
%change zarr if the code take too long to run
einter = zeros(length(zarr),length(rarrall)); %array of field of
multimode interference
einter0 = zeros(1,length(rarrall)); %array of field of multimode
interference at z = 0
for nn = 1:length(zarr)
    for mm = 1:length(rarrall)
        for pp = 1:length(neffarrmmf)
            einter(nn,mm) = einter(nn,mm) +
(aj(pp)*earrallmmf(pp,mm)*exp(1j*betammf(pp)*zarr(nn)));
        end
    end
end
for mm = 1:length(rarrall)
    for pp = 1:length(neffarrmmf)
        einter0(mm) = einter0(mm) +
(aj(pp)*earrallmmf(pp,mm)*exp(1j*betammf(pp)*0));
    end
end
iinter = (abs(einter)).^2; %array of field intensity of
multimode interference
iinter0 = (abs(einter0)).^2; %array of field intensity of
multimode interference at z = 0
%
contourf(rarrall,zarr,iinter,'LineStyle','none','LevelList',0:0.
01:1)
% hold on
% colormap(jet)

%-----------------------finding coupling efficiency of mmf with
interference----------------------------------%
funescore = @(r) ((abs(besselj(0,(usmf.*r./asmf)))).^2).*r;
funesclad = @(r) ((abs(c*besselk(0,(wsmf.*r./asmf)))).^2).*r;
funesair = @(r) ((ecladdingsmf2(length(ecladdingsmf2))).^2).*r;
% ps = (integral(funescore,0,asmf) +
integral(funesclad,asmf,ammf) +
integral(funesair,ammf,Inf)).*(2*pi);
ps = (integral(funescore,0,asmf) + integral(funesclad,asmf,ammf)
+ 0).*(2*pi); %assume third integral is zero
ajbar = aj.*(sqrt(pj./ps)); %aj bar parameters of each mode
```

```
etaarr = zeros(size(zarr));
for nn = 1:length(zarr)
    for mm = 1:length(neffarrmmf)
        beth = betammf(mm);
        ahb = ajbar(mm);
        for pp = 1:length(neffarrmmf)
            ajb = ajbar(pp);
            betj = betammf(pp);
            etaarr(nn) = etaarr(nn) +
((ajb^2)*((conj(ahb))^2)*exp(1j*(betj - beth)*zarr(nn)));
        end
    end
end
plot(zarr,10.*log(etaarr)./log(10))
```

ii. Characteristic equation for singlemode fibre (characeqnsmf)

```
function f = characeqnsmf(neff)
u = findusmf(neff);
w = findwsmf(neff);
f = ((besselj(0,u))*(w)*(besselk(1,w))) -
((besselk(0,w))*(u)*(besselj(1,u)));
end
```

iii. Function of finding parameter u for singlemode fibre (findusmf)

```
function f = findusmf(neff)
global asmf k0 epscoresmf
f = k0.*asmf.*(sqrt(epscoresmf - (neff.^2)));
end
```

iv. Function of finding parameter w for singlemode fibre (findwsmf)

```
function f = findwsmf(neff)
global k0 asmf epscladdingsmf
f = k0.*asmf.*(sqrt((neff.^2) - epscladdingsmf));
end
```

v. Characteristic equation for multimode fibre (characeqnmmf)

```
function f = characeqnmmf(neff)
u = findummf(neff);
w = findwmmf(neff);
f = ((besselj(0,u)).*(w).*(besselk(1,w,1))) -
((besselk(0,w,1)).*(u).*(besselj(1,u)));
```

```
end
```

vi. Function of finding parameter u for multimode fibre (findummf)

```
function f = findummf(neff)
global ammf k0 epscoremmf
f = k0.*ammf.*(sqrt(epscoremmf - (neff.^2)));
end
```

vii. Funciton of finding parameter w for multimode fibre (findwmmf)

```
function f = findwmmf(neff)
global k0 ammf epscladdingmmf
f = k0.*ammf.*(sqrt((neff.^2) - epscladdingmmf));
end
```