Ⅰ. 서론

1.1 테라헤르츠

테라헤르츠 (Terahertz, THz) 파는 마이크로파와 적외선 사이에 존재하는 0.1 ~ 10 THz 전자기파를 의미한다. 파장으로는 30 ~ 3000 μm, 광자 에너지로는 0.41 ~ 41 meV의 물리량을 가진다. 테라헤르츠파는 전자공학과 광공학의 경계에 위치하기 때문에 융합형 과학기술로의 발전 가능성이 크다.

최근 몇 년간 무선 통신의 데이터 전송량이 기하급수적으로 증가하고 있다 [1]. 이는 “필요한 데이터 전송량 18개월마다 2배로 증가한다”는 Edholm’s law에 의해서도 예측된 상황이기도 하다 [2]. 따라서 데이터 전송량이 늘어나면서 ‘beyond 5G’ 시대에서는 더 넓은 대역의 주파수가 필요해졌다. 이에 테라헤르츠가 가장 실현 가능성 있고 적합하다고 논의되고 있다 [3,4]. 하지만 테라헤르츠는 무선 환경에서 수증기에 흡수되어 손실되는 신호가 많기 때문에, 아주 먼 거리에서의 무선 통신은 어렵다 [5]. 따라서 테라헤르츠 광통신 (fiber optic communication) 연구가 우선적으로 진행되어야 한다.



그림 1.1 전자기파의 주파수 대역 스펙트럼

1.2 머신 러닝

테라헤르츠 광통신을 위해서는 테라헤르츠 브레그 분산 반사경 (Distributed Bragg Reflector, DBR)과 광전력 분배기 (Optical Power Splitter)가 필수적이다. DBR 같은 경우 이론적인 모델이 정립되어 있어, 높은 효율의 소자를 만드는데 어려움이 없다. 그러나 광전력 분배기 같은 경우 이론적으로 해석하는데 어려움이 있기 높은 효율을 가지도록 설계하는데 어려움이 있다. 기존에는 직접 이진 검색 (Direct Binary Search, DBS) [6,7], 기본 구조에서 길이나 광학 상수들을 변수로 하여 각각의 변수에 해당하는 모든 소자의 성능을 계산하고 비교하여 최적의 구조를 찾는 방법 [8,9], 일정 주기로 반복되는 패턴을 이용한 자기 이미징 (Self-Imaging) 방법 [10,11] 등으로 설계를 하였다. 그러나 이 방법들은 필요한 계산 장비 성능도 높았고, 계산 시간도 오래 걸린다. 또한 이러한 방식으로 설계된 구조들은 실현 가능성 (Design feasibly)과 스펙트럼 복잡성 (Spectral complexity)이 반비례하기 때문에 원하는 스펙트럼을 가진 실현 가능한 구조를 찾기 어렵다 [12]. 머신 러닝 (Machine Learning)으로 광학 소자를 설계한다면 이러한 한계들을 극복할 수 있다.



그림 1.2 광학 소자 설계 방법에 따른 실현 가능성과 스펙트럼 복잡성의 관계

머신 러닝은 인공 지능 (Artificial Intelligence)의 한 분야로 어떠한 문제 (tasks)로부터 나오는 결과 (performance measure)로부터 데이터 (experience)를 얻어 학습을 하는 프로그램을 말한다 [13]. 실제 적용시킬 때 이 데이터들을 학습 데이터 (training data)라고 한다. 딥 러닝은 머신 러닝의 한 분야로 학습이라는 기본 방식은 같지만, 문제를 기본 개념 (simple concepts)으로부터 추가적인 개념 (nested hierarchy of concepts)을 스스로 생성해 더 쉽게 문제를 해결한다. 예를 들어 사진을 보고 개와 고양이를 구분하는 문제를 머신 러닝과 딥 러닝으로 해결한다고 생각한다면, 머신 러닝은 문제를 해결하기 위해서 개와 고양이의 특징들 (features)을 프로그램에 입력해주어야 한다. 이러한 경우, 개와 고양이 각각의 종마다 생김새가 다르기 때문에 많은 특징들을 생각해야 한다는 어려움이 있다. 하지만 딥 러닝은 학습 데이터들로부터 스스로 특징들을 생성하기 때문에 좀 더 편리하게 문제를 해결 할 수 있다. 머신 러닝은 세 분류로 나눌 수 있다: 지도 학습 (Supervised Learning), 비지도 학습 (Unsupervised Learning), 강화 학습 (Reinforcement Learning). 지도 학습은 데이터가 어떠한 것인지 명시가 돼있는 상태에서 학습하는 방법이다. 비지도 학습은 데이터가 어떠한 것인지 명시 되지 않은 상태에서 학습하는 방법이다. 강화 학습은 문제 상태에 따라 프로그램이 어떠한 행동을 하도록 하는데, 그에 해당하는 보상(reward)를 주면서 보상을 최대화하는 행동을 하도록 학습하는 방법이다.

본 논문에서는 테라헤르츠 광통신 연구를 위한 기초연구로 1 테라헤르츠 (1 THz)에서 동작하는 DBR과 1×2 광전력 분배기를 설계하였다. 먼저 DBR과 1×2 광전력 분배기에 대해 알아보고, 머신 러닝을 이용하여 설계하기 적합하도록 문제 상황을 설계하였다. 그 후 머신 러닝에 속하는 가산 강화 학습 방법과 딥 러닝에 속하는 인공 신경망 방법을 이용하여 DBR과 1×2 광전력 분배기를 설계하였다.

1.3 머신 러닝 도구 및 전산 자원

본 연구에서는 머신 러닝을 Anaconda[14] 가상 환경에서 프로그래밍 언어 파이썬 (Python)을 사용하였다. 인공 신경망을 위해서 구글의 오픈 소스인 Tensorflow[15]를 사용하였다. 전산 자원은 다음과 같다.

Ⅱ. 분산 브레그 반사경

2.1 다층 구조에서의 전달 행렬(Transfer Matrix)]

DBR과 같은 다층 구조에서 빛의 전파 특성은 전달 행렬을 사용하여 쉽게 계산할 수 있다 [16]. 그림 2.1과 같이 매질이 x 방향에 대해 균일하지 않다면, 유전율(permittivity, ε)과 투자율(permeability, μ)을 구간에 대해 나누어 생각할 수 있다.

for



그림 2. 1 다층 구조에서의 빛의 전파

입사하는 빛과 반사하는 빛이 모두 TE mode (Transverse Electric mode)라고 생각하면 각각의 전기장(**E**i, **E**r)은 다음과 같이 나타낼 수 있다.

*l* 번째 매질 ()에서의 전기장은 이며, 자세히 나타내면 다음과 같다.

이때 , 이다. 전기장의 탄젠트(tangent) 성분 *Ey*와 자기장의 탄젠트 성분 *Hz*가 매질의 경계인 에서 연속이라는 경계 조건을 사용하면 다음 식을 얻을 수 있다.

위 두 식을 연립하여 정리하면 두 매질에서 빛의 전달 관계식을 행렬 형태로 나타낼 수 있게 된다. 이를 좀더 간단히 나타내기 위해서 다음과 같이 *Pl(l+1)*와 *l*+1번째 매질의 두께를 정의한다.

또한, *l*+1번째 매질의 두께를 다음과 같이 나타낼 수 있다.

위의 두 식을 정리하면 다음과 같이 전달 행렬을 나타낼 수 있다.

여기서, 은 *l*+1번째 매질에서 *l*번째 매질로 진행하는 빛을 계산하기 때문에 역방향 전파 행렬(backward-propagation matrix)라고 한다. 정방향 전파 행렬(forward-propagation matrix) 도 다음과 같이 구할 수 있다.

앞에서 구한 전달 행렬을 N+1 개의 층을 가진 다층 구조에 적용하면 전체 구조에 대한 빛의 전파 특성을 다음과 같이 구할 수 있다.

이로부터 다층 구조에 대한 투과 계수(transmission coefficient) t와 반사 계수(reflection coefficient) r를 다음과 같이 구할 수 있다.

,

2.2 분산 브레그 반사경 (Distributed Bragg Reflector)의 구조와 특성

Distributed Bragg Reflector (DBR)은 서로 다른 굴절률(index, n)을 가지는 두 물질을 번갈아 배치한 다층 구조의 반사 장치이다. 일반적으로 DBR 양쪽 끝에 같은 물질을 사용하기 때문에 N+1/2 pairs 구조를 가진다. DBR 각 매질의 광학적 두께 (optical thickness)는 반사를 위한 공진 조건을 만족하려면 설계 파장 (design wavelength)의 1/4이 되어야 한다. 전달 행렬을 N+1/2 pairs DBR 구조에 적용하면 다음과 같다.

여기서 아래 첨자 0은 빛이 입사하는 매질, s는 빛이 전체 구조를 투과한 이후의 매질을 나타내며 보통 같은 매질이다. *h*와 *l*은 각각 상대적으로 높고 낮은 굴절률을 가지는 매질을 의미한다.



그림 2. 2 N+1/2 paris의 주기적인 DBR 구조

2.3 DBR 전산모사 도구

머신 러닝을 이용하여 DBR을 설계하기 위해서는 무작위 다층 구조로 이루어진 학습 데이터가 필요하다. 이는 2.1에서 서술한 전달 행렬을 계산하여 만들 수 있다. 전산 모사 소프트웨어로는 Python에서 오픈 소스(open source)인 TMM(Transfer Matrix Method) package [17]를 사용하였다.

Ⅲ. 1×2 광전력 분배기

3.1 1×2 광전력 분배기 (Optical Power Splitter)

1×2 Optical Power Splitter (OPS)는 하나의 도파관 (waveguide)에 두개의 도파관을 Y 자로 나뭇가지처럼 (Y-branch) 연결한 형태로, 하나의 입력된 광을 두 개의 똑같은 전력을 가진 광으로 분배하는 소자이다. 광통신 시스템에서 효율적인 광전송을 위해서는 광섬유 사이간의 효율적인 결합이 필요하기 때문에 OPS에 대한 연구가 많이 진행되고 있다 [18, 19].



그림 3. 1 Y-branch 형태의 OPS

기본 OPS 형태인 Y-branch는 branch angle이 작을수록 branch 입력에서 발생하는 분산 손실 (scattering loss)과 출력에서 발생하는 전환 손실 (conversion loss)을 줄일 수 있다 [20]. 따라서 branch distance가 도파관 두께에 비해 길게 설계하게 되는데, 이 때문에 소자의 크기가 불필요하게 증가하게 된다. 머신 러닝을 이용하여 OPS를 설계할 경우에는 이 문제를 해결하기 위해 그림 3.2와 같은 구조를 이용한다.



그림 3. 2 머신 러닝 설계를 위해 변경된 OPS 구조

그림 3.2 (a)-1과 같이 기존 Y-branch 구조에서 설계 영역을 20 × 20 pixels로 나누어 준다. 이때 설계 목표에 맞도록 pixels의 매질을 결정하게 된다. 그림 3.2 (b)-1처럼 정사각형의 pixel을 그대로 사용하게 되면, 실제 공정에서 직각을 나타내기가 어려워 공정 효율이 떨어지게 된다. 따라서 그림 3.2 (b)-2와 같이 정사각형 pixel을 원형 pixel로 바꾸어, 같은 효과를 보이면서도 공정 효율을 획기적으로 증가시킬 수 있다 [6].

1×2 OPS의 목표는 하나의 입력된 광을 두 개의 똑같은 전력을 가진 광으로 분배하는 것이기 때문에 대칭 성질을 사용한다. 대칭 성질을 사용하면 전산모사 소요시간이 감소하며, 설계 난이도도 감소하게 된다. 따라서 1×2 OPS의 성능 검사 지표는 두 출력의 전력 유사도가 아니라, 내부 손실 (insertion loss)과 전력 균일도 (power uniformity)이다. 이 때 내부 손실은 입력 대비이기 때문에, 1×2 OPS의 이상적인 내부 손실값은 3dB이다.

3.2 1×2 OPS 전산모사 도구

OPS의 branch를 계단 형태로 근사시켜 이론적으로 해석할 수도 있다. 하지만, 본 논문에서 설계하는 OPS는 QR code-like 구조이기 때문에 전산모사를 통해 해석한다. 전산모사는 Finite-Difference Time-Domain (FDTD) 방법을 사용하였다. FDTD 전산모사는 대상을 아주 작은 사각형 cell로 나누어 Maxwell 방정식으로 근사없이 정확하게 계산하는 도구이다. 따라서 복잡한 구조에서도 높은 정확도를 보인다. 전산모사 소프트웨어는 Lumerical 사의 상용 소프트웨어인 FDTD Solutions [21]를 사용하였다.

Ⅳ. 가법 강화 학습

4.1 퍼셉트론(Perceptron)

퍼셉트론은 1957년 Frank Rosenblatt에 의해 제안된 초기 형태의 인공 신경망이다 [22]. 다수의 입력으로부터 하나의 결과를 내보내는 알고리즘으로 실제 뇌를 구성하는 뉴런 (neuron)과 유사한 동작을 한다.



그림 4. 1 퍼셉트론

그림 4.1에서 x는 입력값, w는 가중치(weight), y는 출력값을 의미한다. x에 입력이 보내지면 w와 곱해진다 (x∙w). 곱해진 값이 바로 y가 되는 것이 아니라, 활성화 함수 (activation function)라는 것을 거치게 된다. 뉴런은 ‘예’ 혹은 ‘아니오’의 동작하기 때문에 y 값도 ‘1’ 또는 ‘0’만 가져야 한다. 하지만 x∙w는 0과 1사이의 실수값이 나오게 되는데, 이 때 필요한 것이 활성화 함수이다. 그래서 퍼셉트론에서는 주로 일정 값 이상이면 1을 출력하고, 미만이면 0을 출력하는 계단 함수 (Step function)를 활성화 함수로 사용한다. 결과적으로, 퍼셉트론은 데이터들의 선형 분류기로 동작을 한다.

4.2 강화 학습(Reinforcement Learning)

강화 학습은 머신 러닝의 한 분야로 어떠한 문제를 해결할 때 컴퓨터가 스스로 최적의 결정을 내릴 수 있도록 학습시키는 방법이다 [23]. 강화 학습 방법으로 문제를 해결하기 위해서는, 그 문제를 Markov Decision Process (MDP)로 정의해야한다. MDP는 다음과 같이 정의된다.

**Definition 4.1** MDP는 5개의 요소(*S*, *A*, *T*, *R*, γ)로 이루어진 집합이다. 각각의 요소는 다음과 같다:

* *S*: 상태 (state)들의 집합
* *A*: 행동 (action)들의 집합
* *T*: 전이 함수 (transition function), 특정 상태에서 특정 행동을 했을 때 다음 상태는 어떤 상태가 될지에 관한 확률
* *R*: 보상 함수 (reward function), 주어진 상태에서 행동을 했을 때 얻게 되는 보상 (reward)
* γ: 감쇠 상수 (discount factor), 처음 받은 보상의 영향을 증가시키고, 시간이 지난 보상의 영향을 줄이는 역할

초기 상태 s0에서 초기 행동 a0를 수행하면 주어진 Ts0a0에 따라 다음 상태 s1이 확률적으로 결정된다. 그 결과로 보상 R(s0, a0)를 얻게 된다. 이 과정을 s1에도 적용시키며 목표 상태에 도달할 때까지 반복한다. 강화 학습의 목표는 시간이 흘렀을 때 얻게 되는 모든 보상들의 합을 최대화 시키는 정책 (policy, π)을 학습하는 것이다. 정책은 상태에서 어떠한 행동을 할지 결정하는 함수이다. 이때 최대화 하고자 하는 보상들의 합은 감쇠 상수와 함께 다음과 같이 정의 된다.

T에 의해 상태에 따른 행동이 정해지기 때문에 실제 최대화 시키는 값은 보상들의 합의 기대값이다.

4.3 가법 강화 학습(Additive Reinforcement Learning)을 이용한 설계

4.3.1 가법 강화 학습

Additive Reinforcement Learning (ARL)은 퍼셉트론의 가법 성질 (additive)과 강화 학습 (Reinforcement)의 보상 시스템을 합한 알고리즘이다 [24]. ARL은 학습 단계 (Training phase)와 추론 단계 (Inference phase) 로 나뉘어 진다 [25]. 학습 단계에서는 데이터로 주어진 구조들의 보상을 계산하여 각 구조들의 설계 목표에 부합하는 정도를 계산한다. *A* 를 우리가 원하는 구조라 가정했을 때, 임의의 구조 *B*에 대해 *A*와 비슷한 정도를 보상으로 나타낼 수 있으며, 그 함수를 *f*(*A*, *B*)라 하자. 보상 *r*을 이진 행렬로 표현한 *B*에 곱하여 보상 총합 행렬 (reward summation matrix, Ɛ)에 누적 시킨다. 보상 총합 행렬은 0으로 초기화 되어있는 상태이다.

이때 구조의 길이를 n, 구조의 개수 (학습 데이터의 개수)를 N이라 하면 다음과 같이 의사 코드 (pseudo code)로 나타낼 수 있다.

**Algorithm 4.1** ARL: Training Phase

1: **for all** *i* ∈ {1, …, *n* \* N} **do**

2: Randomly create *n* × *n* binary guess matrix *B*

3: Get *r* = *f*(*A*, *B*)

4: Ɛ ← Ɛ + *r* \* *B*

5: **end for**

실제 광학 소자를 설계 할 때는 원하는 구조 *A*를 모르는 상태이기 때문에 설계자가 보상 함수가 *f*(*B*)를 정의하여야 한다. 예를 들어, DBR을 설계하고자 할 때는 구조 *B*의 반사율 (reflectance)을 보상 함수로 정의하여 반사율이 최대가 되는 구조를 찾는다.

추론 단계에서는 학습 단계에서 계산한 reward summation matrix, Ɛ로부터 최종 결과물을 얻는다. Ɛ의 최소값을 구하여 모든 원소에서 빼준다.

그리고 Ɛ 모든 원소의 평균을 *avg*라 하고, 모든 원소가 0으로 초기화된 상태의 최종 결과를 저장할 행렬을 다음과 같이 *F*라 한다.

그 다음 Ɛ*i*, *j*가 *avg*보다 크거나 같으면 *Fi*, *j*에 1을 작으면 0을 대입해준다. 다음과 같이 의사 코드로 나타낼 수 있다.

**Algorithm 4.2** ARL: Inference Phase

1: **for all** *i* ∈ {1, …, *n* } **do**

2: **for all** *j* ∈ {1, …, *n* } **do**

3: **if** Ɛ*i, j* ≥ *avg* **then**

4: *Fi, j* = 1

5: **else**

6: *Fi, j* = 0

7: **end if**

8: **end for**

9: **end for**

이 알고리즘의 장점은 실제 코드로 구현이 간편하다는 것이다. 기존 광학 소자 알고리즘들은 광학적 지식뿐만 아니라, 프로그래밍에 대한 높은 이해도 필요하여 일반적으로 사용하기에 무리가 있다. 그러나 이 알고리즘은 충분한 양의 학습 데이터만 있다면 간단한 연산 수준의 프로그래밍만으로 결과를 얻을 수 있다. 단점은 학습 데이터가 결과를 얻기에 충분하려면 다른 알고리즘들 보다 많은 양이 필요하다. 알고리즘 제안 논문 [24]에서 *n*×*n* 이진 행렬을 찾는 문제로 테스트하였을 때, 10,000개 데이터로 *n*이 10이하일때는 90% 정확도로 답을 찾는 것이 가능했다. 하지만 11부터는 정확도가 90% 미만으로 떨어지며 13일 때는 정확도가 50%로 이하로 떨어졌다.



그림 4. 2 행렬 크기 (*n*) 변화에 따른 ARL 알고리즘 정확도

하지만 데이터의 개수가 100,000개 되면 정확도가 모두 100%이 된다. 따라서 *n*이 11, 구조의 변수가 100 이상이 되면 약 100,000개의 학습 데이터가 필요하다.

4.3.2 ARL을 이용한 DBR 설계 방법

DBR의 설계 목표 주파수는 1 THz (= 300 μm) 이며, 실제 제작으로 고려하여 공기와 fused quartz를 사용하였다. 설계에서 고려한 파장 영역은 5 μm 단위로 나누어진 150 ~ 3000 μm 이다. 물질들의 굴절률은 테라헤르츠 대역에서 변화가 거의 없기 때문에 설계 목표 주파수인 1 THz에서의 굴절률로 고정시켰다. 따라서 공기의 굴절률은 1 (*nl* = 1)이며, fused quartz (SiO2)의 굴절률은 2.092 [26]이다 (*nh* = 2.092).



그림 4. 3 ARL을 이용한 DBR 설계 플랫폼

DBR 구조를 이진 행렬 형태로 표현하기 위하여 일정 두께 ∆*x*를 가지는 단위 층 (unit layer)들이 쌓여있는 구조를 생각한다. 이 때 이진 행렬의 값이 1이면 quartz의 굴절률을, 0이면 공기의 굴절률을 대입한다. 이진 행렬의 길이가 *N*이면 실제 DBR 전체 길이는 *N*∙∆*x*가 된다. 공진 조건을 고려하여 DBR 각층의 두께를 구하면 *dh* = 35.85 μm, *dl* = 75 μm 이다. 한 pair의 두께가 약 100 μm 이고, 3.5 ~ 4.5 pairs 정도의 두께를 고려하여 L = 400 μm로 설정하였다. ∆*x*가 너무 작으면 N이 증가하기 때문에 ARL에 필요한 데이터의 개수가 증가하고, 너무 크게 되면 정확한 층 두께를 구할 수 없게 된다. 따라서 공진 조건을 고려하여 구한 두께를 참고하여 ∆*x* = 5 μm, *N* = 80으로 설정하였다. 설계에서 고려한 빛의 진행 mode는 TE mode이다.

설계 주파수에서 대역폭을 정의하고 그림 4.4에서 보이듯이 대역폭에 해당하는 반사율을 *R*in, 해당하지 않는 반사율을 *R*out이라 하고 다음과 같이 보상 *r*을 정의하였다.

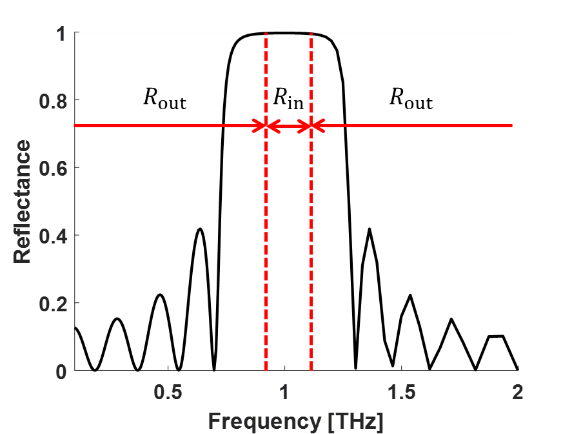
****

그림 4. 4 ARL을 이용한 DBR 설계 보상 정의 방법

학습 데이터로 길이 *N*의 이진 행렬을 무작위로 생성하여 총 20,000개의 구조와 반사율을 구하였다. 데이터 분포를 보상에 대해 정리하면 그림 4.5로 나타낼 수 있다.



그림 4. 5 ARL을 이용한 DBR 설계 학습 데이터 분포도

4.3.2 ARL을 이용한 DBR 설계 결과

그림 4.6은 ARL을 이용하여 설계한 DBR과 공진 구조를 고려하여 설계한 DBR의 반사율을 비교한 그래프이다. ARL을 이용한 설계에서 전산모사를 파장 단위로 계산하여 설계 파장을 300 μm, 대역폭을 100 μm로 하여, 주파수 단위로 변환한 그래프에서는 양끝이 1.2 THz, 0.85 THz가 되어 대역폭이 치우쳐져 있는 형태로 나타난다.

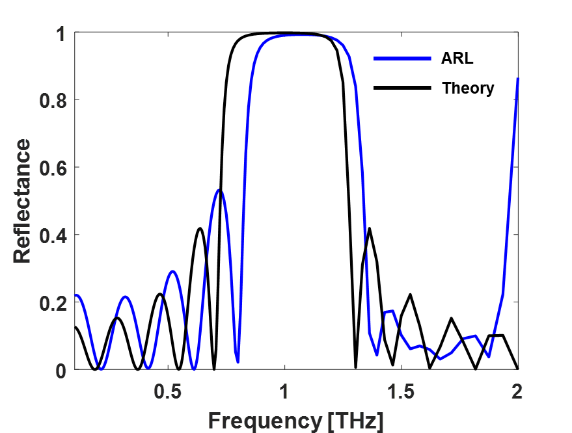


그림 4. 6 ARL을 이용하여 설계한 DBR과 이론으로 설계한 DBR의 반사율 비교

|  |  |  |  |
| --- | --- | --- | --- |
|  | Maximum Reflectance of DBR | FWHM | 99% bandwidth |
| ARL | 0.9928 at  1.07 THz (280 μm) | 0.541 THz  (145 μm) | 0.287 THz  (75 μm) |
| Theory | 0.9978 at  0.99 THz (300 μm) | 0.553 THz  (180 μm) | 0.354 THz  (110 μm) |

표4. 1 ARL을 이용하여 설계한 DBR과 이론으로 설계한 DBR의 반사율 비교



그림 4. 7 ARL을 이용하여 설계한 DBR과 이론으로 설계한 DBR의 두께 비교

ARL을 이용하여 설계한 DBR이 이론으로 설계한 DBR보다 수치적으로 부족한 성능을 보인다. 하지만 설계 주파수 1 THz에서의 반사율 차이는 크지 않으며, 같은 pair이면서도 전체 길이가 공진 조건을 고려하여 설계한 DBR보다 약 100 μm 짧기 때문에 공간 제한 문제가 있는 환경에서 유용하게 사용될 수 있다.

4.3.3 ARL을 이용한 1×2 OPS 설계 방법

1×2 OPS의 설계 목표는 일정 대역폭의 투과율을 최대화시키는 것이 아니라, 원하는 광대역의 투과율을 최대화 시키는 것이다. 목표 주파수가 1 THz이기 때문에 설계 대역폭을 275~325 μm 로 한다. 이를 2 μm 단위로 나누어 총 26개 지점에서 투과율을 측정한다. OPS 구조에서는 식각이 활성화된다는 의미로 ‘1’이 굴절률이 낮은 공기, ‘0’이 굴절률이 높은 실리콘 (Silicon, Si)이다. 이진 행렬로 표현하면 2차원으로 나타내어진다.



그림 4. 8 ARL을 이용한 1×2 OPS 설계 플랫폼

그림 4.7 은 ARL을 이용한 1×2 OPS 설계 플랫폼을 나타낸 것이다. 500 μm×500 μm 정사각 실리콘 위에 9 μm 반지름의 원형 구멍을 식각하여 만든다. Air cladding이기 때문에 원형 구멍의 굴절률은 1이다. 입력과 출력 포트 (port)들은 테이퍼 (taper)가 없으며, 너비가 100 μm인 무한한 길이의 직사각형으로 가정한다. 입력 포트는 정사각 실리콘 중앙에 위치하며, 출력 포트는 각각 중앙으로부터 125 μm 거리에 위치해 있다. OPS는 대칭 구조이기 때문에 T1과 T2의 수치가 같다. 따라서 T로 통일시켜 사용한다. 입력으로는 FDTD에서 기본 모드 (fundamental mode)로 계산된 TM mode (Transverse Magnetic mode)를 사용하였다.

설게 목표가 광대역의 투과율을 최대화 시키는 것이기 때문에 보상을 다음과 같이 정의한다.

학습 데이터로 20×10 이진 행렬을 무작위로 생성하여 총 100,000개의 구조와 투과율을 구하였다. 데이터 분포를 보상에 대하여 나타내면 그림과 같다.

4.3.4 ARL을 이용한 1×2 OPS 설계 결과

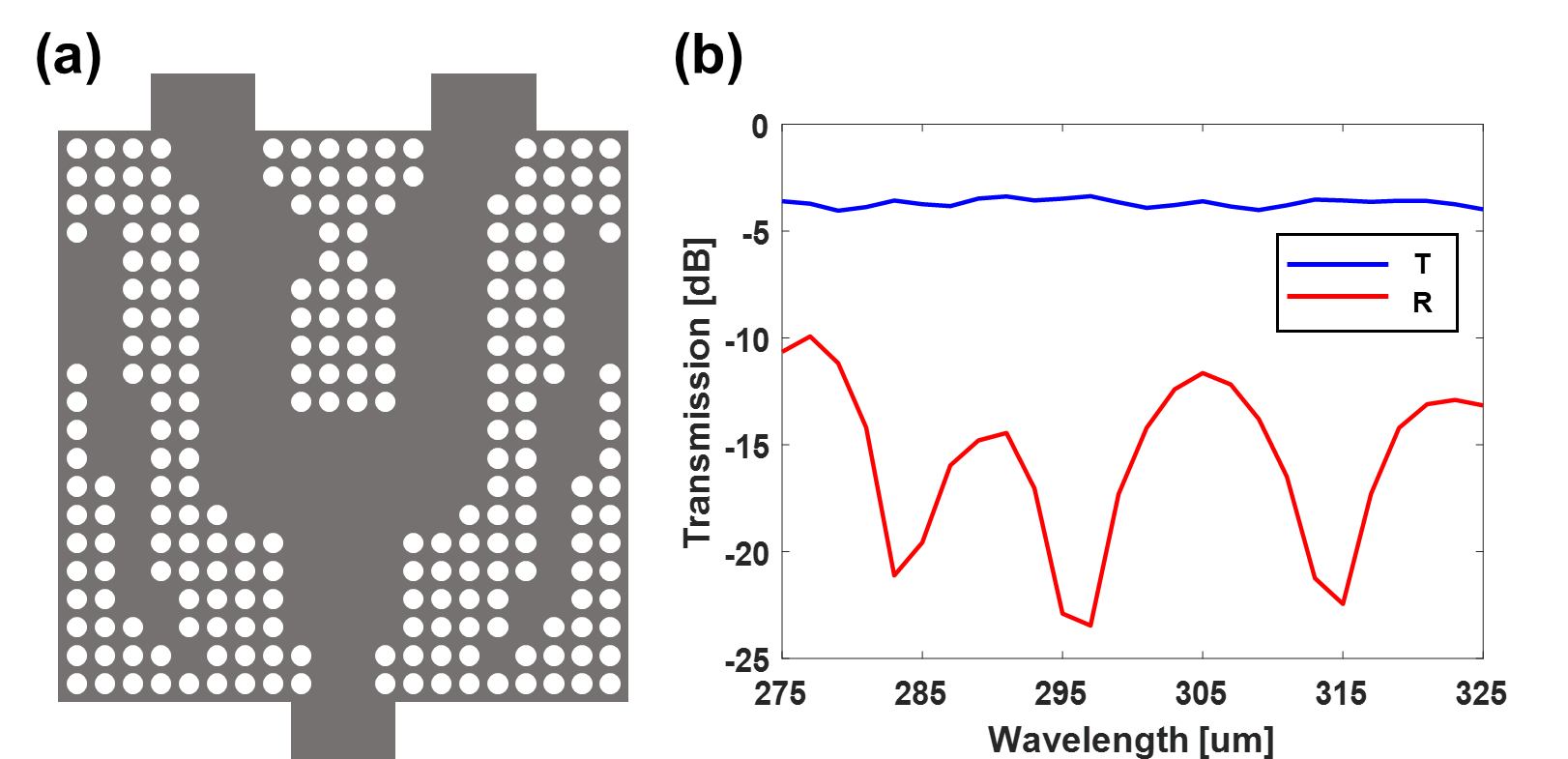


그림 4. 9 ARL을 이용하여 설계한 1×2 OPS (a) 식각 구조 (b) 입력 포트 반사율 (R)과 출력 포트 (T) 투과율

ARL을 이용하여 설계한 OPS의 내부 손실은 평균 3.69 dB로 4 dB보다 적다. 또한, 균일도는 0.68 dB로 준수한 수준이다.

Ⅴ. 인공 신경망

5.1 인공 신경망(Artificial Neural Network)

5.1.1 인공 신경망의 구조와 역할

인공 신경망은 여러 개의 퍼셉트론 노드로 이루어진 층 (layer)과 이런 층들이 이어져 다층 구조 (multi-layer)로 이루어진 형태이다. 학습 데이터를 입력하는 layer를 input layer라 하고, 연산 결과가 출력되는 layer를 output layer라 한다. 그 사이에 있는 layer들은 실질적으로 보이지 않기 때문에 hidden layer라 한다. 각 노드들에는 가중치가 있으면, 어떠한 입력을 주었을 때 우리가 원하는 출력이 나오도록 가중치를 조절하는 과정을 학습 (learning)이라 한다. 학습은 입력과 출력이 pair로 이루어져있는 학습 데이터들을 이용하여 이루어진다.

학습 데이터의 *i*번째 입력을 *xi*, 출력을 *yi*라 하면, 초기 학습되지 않은 상태의 인공 신경망에 *xi*를 입력하면 가 출련된다. 이때 *yi*와 의 차이, cost를 다음과 같이 정의한다.

계산되는 cost를 줄이기 위해 노드들의 가중치를 조절하는 것이 optimizer이다. 가장 많이 사용되는 optimizer는 Adam optimizer이다. Optimizer가 cost를 계속해서 최소화 시키는 과정을 학습이라고 한다. 그래서 학습이 진행되는 과정을 step에 따라 변화하는 cost를 시각화하여 확인할 수 있다.

5.1.2 오버피팅(Overfitting)

인공 신경망을 학습시킬 때 단순히 많은 양의 학습 데이터를 제공한다 하여도 정확한 결과가 나타나지 않을 수 있다.

다양한 분포를 가지는 데이터 세트를 사용하거나, 인공 신경망의 구조를 변경하는 것이 일반적인 해결법이다.

5.2 인버스 디자인(Inverse Design)

인버스 디자인은

5.3 인공 신경망을 이용한 인버스 디자인

5.3.1 인공 신경망을 이용한 인버스 디자인 방법

기존 인버스 디자인들은 프로그래밍과 물리 이론들에 대한 깊은 이해가 필요하여 광학 소자를 설계함에 있어 일반적인 방법으로 쓰이기에는 어려움이 많다. 그 원인은 구조의 response를 계산하는 부분의 구현과 이의 inverse를 구하는 과정이다. 인공 신경망은 간단한 연산과정이기 때문에 inverse를 어렵지 않게 구할 수 있다. 따라서 구조의 response를 predict하는 인공 신경망을 만든다.

5.3.2 인공 신경망을 이용한 DBR 인버스 디자인

5.3.3 인공 신경망을 이용한 1×2 OPS 인버스 디자인

Ⅵ. 결론

최근 통신 분야에서 테라헤르츠가 각광받고 있다. 데이터 전송량이 늘어나면서 더 넓은 대역의 주파수가 필요해졌기 때문이다. 하지만 테라헤르츠는 수증기에 흡수되는 양이 많기 때문에 아주 먼거리에서의 무선 통신은 어렵다. 따라서 광케이블을 이용한 테라헤르츠 광케이블 연구가 진행되어야 한다.

**REFERENCES**

1. **F. Akyildiz, J. M. Jornet, and C. Han, “Terahertz band: Next frontier for wireless communications,” Physical Communication, vol. 12, 2014, pp. 16–32.**
2. **S. Cherry, “Edholms law of bandwidth,” IEEE Spectrum, vol. 41, no. 7, 2004, pp. 58–60.**
3. **J. Federici and L. Moeller, “Review of terahertz and subterahertz wireless communications,” Journal of Applied Physics, vol. 107, no. 11, 2010, p. 111101.**
4. **T. Kleine-Ostmann and T. Nagatsuma, “A Review on Terahertz Communications Research,” Journal of Infrared, Millimeter, and Terahertz Waves, vol. 32, no. 2, 2011, pp. 143–171.**
5. **H.-J. Song and T. Nagatsuma, “Present and Future of Terahertz Communications,” IEEE Transactions on Terahertz Science and Technology, vol. 1, no. 1, 2011, pp. 256–263.**
6. **L. Lu, M. Zhang, F. Zhou, and D. Liu, “An Ultra-compact Colorless 50:50 Coupler Based on PhC-like Metamaterial Structure,” Optical Fiber Communication Conference, 2016.**
7. **K. Chandu, M. Stanich, C. W. Wu, and B. Trager, “Direct binary search (DBS) algorithm with constraints,” Color Imaging XVIII: Displaying, Processing, Hardcopy, and Applications, 2013.**
8. **P. Besse, E. Gini, M. Bachmann, and H. Melchior, “New 2×2 and 1×3 multimode interference couplers with free selection of power splitting ratios,” Journal of Lightwave Technology, vol. 14, no. 10, 1996, pp. 2286–2293.**
9. **Y. Tian, J. Qiu, M. Yu, Z. Huang, Y. Qiao, Z. Dong, and J. Wu, “Broadband 1 × 3 Couplers With Variable Splitting Ratio Using Cascaded Step-Size MMI,” IEEE Photonics Journal, vol. 10, no. 3, 2018, pp. 1–8.**
10. **M. Bachmann, P. A. Besse, and H. Melchior, “General self-imaging properties in N × N multimode interference couplers including phase relations,” Applied Optics, vol. 33, no. 18, 1994, p. 3905.**
11. **L. Soldano and E. Pennings, “Optical multi-mode interference devices based on self-imaging: principles and applications,” Journal of Lightwave Technology, vol. 13, no. 4, 1995, pp. 615–627.**
12. **I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, “Plasmonic nanostructure design and characterization via Deep Learning,” Light: Science & Applications, vol. 7, 2018, no. 1.**
13. **T. M. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.**
14. **Anaconda Software Distribution. Computer software. Vers. 2-2.4.0. Anaconda, Nov. 2016. Web. <https://anaconda.com>.**
15. **Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” arXiv preprint arXiv:1603.04467, 2016.**
16. **S. L. Chuang, Physics of optoelectronic devices. New York, NY: Wiley, 1995.**
17. **Steven J. Byrnes, “Multilayer optical calculations,”** **arXiv preprint arXiv:1603.02720v3 [physics.comp-ph]**
18. **L. Han, B. P.-P. Kuo, N. Alic, and S. Radic, “Ultra-broadband multimode 3dB optical power splitter using an adiabatic coupler and a Y-branch,” Optics Express, vol. 26, no. 11, 2018, p. 14800.**
19. **F. Costache, H. Hartwig, K. Bornhorst, and M. Blasl, “Variable Optical Power Splitter with Field-Induced Waveguides in Liquid Crystals in Paranematic Phase,” Optical Fiber Communication Conference, 2014.**
20. **W. S. C. Chang, Fundamentals of guided-wave optoelectronic devices. Cambridge: Cambridge University Press, 2010.**
21. **Lumerical Inc. https://www.lumerical.com/products/**
22. **F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain.,” Psychological Review, vol. 65, no. 6, 1958, pp. 386–408.**
23. **V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, “An Introduction to Deep Reinforcement Learning,” Foundations and Trends® in Machine Learning, vol. 11, no. 3-4, 2018, pp. 219–354.**
24. **Çağrı Latifoğlu, “Binary Matrix Guessing Problem,” arXiv preprint arXiv:1701.06167v2, 2017.**
25. **M. Turduev, E. Bor, C. Latifoglu, I. H. Giden, Y. S. Hanay, and H. Kurt, “Ultracompact Photonic Structure Design for Strong Light Confinement and Coupling Into Nanowaveguide,” Journal of Lightwave Technology, vol. 36, no. 14, 2018, pp. 2812–2819.**
26. **C. L. Davies, J. B. Patel, C. Q. Xia, L. M. Herz, and M. B. Johnston, “Temperature-Dependent Refractive Index of Quartz at Terahertz Frequencies,” Journal of Infrared, Millimeter, and Terahertz Waves, vol. 39, no. 12, 2018, pp. 1236–1248.**