Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – **Innova Lee**(이상훈)

gcccompil3r@gmail.com

학생 hyungjun Yu(유형준)

love592946@nave.rcom

0516 숙제

벡터의 기본개념 익히기

1) 벡터의 개념

(1) 스칼라와 벡터

스칼라량과 벡터량

- 스칼라량 : 크기만으로 표현하는 양 $A,\ a,\ \overline{A},\ \overline{OP}$
- 벡 터 량 : 크기와 방향으로 표현하는 양 $\mathbf{A},\ \mathbf{a},\ \vec{A},\ \overrightarrow{OP}$

◉ **스칼라량과 벡터량의 예**

벡터의 내적(스칼라 곱)

$$\vec{a}\circ\vec{b} = |\vec{a}||\vec{b}|\cos\theta$$

(단, θ는 주어진 두 벡터의 사이각)

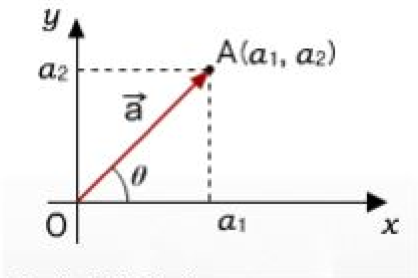
벡터의 외적(벡터 곱)

$$\vec{a}\times\vec{b} = \vec{n}|\vec{a}\times\vec{b}| = \vec{n}|\vec{a}||\vec{b}|\sin\theta$$

(단, $\vec{n}$ 은 주어진 두 벡터에 수직인 단위 벡터
θ는 주어진 두 벡터의 사이각)

(2) 벡터의 성분

◉ 평면 벡터의 성분



위치벡터 $\vec{a}(=\overrightarrow{OA})$는 원점을 시점으로 하고, (a_1, a_2)를 종점으로 하므로
좌표평면에 있는 하나의 점과 대응된다.
따라서 다음과 같이 나타낼 수 있다.
$\vec{a}=(a_1, a_2)$ 또는 $\overrightarrow{OA}=(a_1, a_2)$

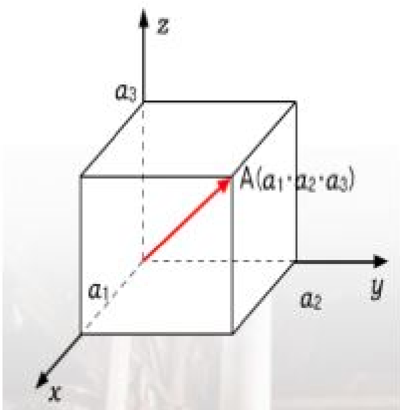
◉ 벡터의 성질

1. 벡터 $\vec{a}$의 크기는 $\overline{OA}$의 길이와 같다. $\therefore |\vec{a}|=\sqrt{a_1^2+a_2^2}$

2. $\tan\theta=\dfrac{\sin\theta}{\cos\theta}=\dfrac{a_2}{a_1}$, $a_1=|\vec{a}|\cos\theta$, $a_2=|\vec{a}|\sin\theta$

3. $\vec{a}=(a_1, a_2)$, $\vec{b}=(b_1, b_2)$라 할 때,
   $\vec{a}=\vec{b} \Leftrightarrow a_1=b_1,\ a_2=b_2$
   $\vec{a}+\vec{b}=(a_1+b_1, a_2+b_2)$, $\vec{a}-\vec{b}=(a_1-b_1, a_2-b_2)$
   $k\vec{a}=(ka_1,\ ka_2)$

◉ 공간 벡터의 성분



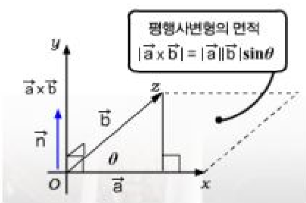
$\vec{a}=(a_1, a_2, a_3)$, $\vec{b}=(b_1, b_2, b_3)$라 할 때,

1. 벡터 $\vec{a}$의 크기
   $|\vec{a}|=\sqrt{a_1^2+a_2^2+a_3^2}$
2. 두 벡터가 서로 같을 조건
   $\vec{a}=\vec{b} \Leftrightarrow a_1=b_1,\ a_2=b_2,\ a_3=b_3$
3. 벡터의 덧셈
   $\vec{a}+\vec{b}=(a_1+b_1, a_2+b_2, a_3+b_3)$
4. 벡터의 뺄셈
   $\vec{a}-\vec{b}=(a_1-b_1, a_2-b_2, a_3-b_3)$
5. 벡터의 실수배
   $k\vec{a}=(ka_1,\ ka_2, ka_3)$

(4) 벡터의 외적

◉ 벡터의 외적

$\vec{a}=(a_1, a_2, a_3)$, $\vec{b}=(b_1, b_2, b_3)$일 때, 벡터의 외적을 다음과 같이 정의한다.



벡터의 외적

$\vec{a}\times\vec{b}=\vec{n}|\vec{a}\times\vec{b}|=\vec{n}|\vec{a}||\vec{b}|\sin\Theta$

(단, $\vec{n}$은 주어진 두 벡터에 수직인 단위 벡터)

$\sin\Theta=\dfrac{|\vec{a}\times\vec{b}|}{|\vec{a}||\vec{b}|}$ ($\because$ $|\vec{a}\times\vec{b}|=|\vec{a}||\vec{b}|\sin\Theta$)

◉ 행렬식의 형식으로 외적 구하기

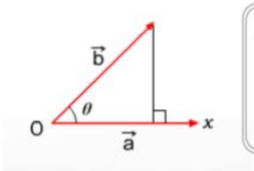
$$|\vec{a}\times\vec{b}|=\begin{vmatrix}\vec{e_1} & \vec{e_2} & \vec{e_3}\\ a_1 & a_2 & a_3\\ b_1 & b_2 & b_3\end{vmatrix}+\begin{vmatrix}a_2 & a_3\\ b_2 & b_3\end{vmatrix}\vec{e_1}-\begin{vmatrix}a_1 & a_3\\ b_1 & b_3\end{vmatrix}\vec{e_2}+\begin{vmatrix}a_1 & a_2\\ b_1 & b_2\end{vmatrix}\vec{e_3}$$

$$=(a_2b_3-a_3b_2)\vec{e_1}+(a_1b_3-a_3b_1)\vec{e_2}+(a_1b_2-a_2b_1)\vec{e_3}$$

(3) 벡터의 내적

◉ 좌표 평면에서의 내적

$\vec{a}=(a_1, a_2)$, $\vec{b}=(b_1, b_2)$일 때, 벡터의 내적을 다음과 같이 정의한다.



벡터의 내적

$\vec{a}\cdot\vec{b}=|\vec{a}||\vec{b}|\cos\Theta$

$=a_1b_1+a_2b_2$

$\cos\Theta=\dfrac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|}=\dfrac{a_1b_1+a_2b_2}{\sqrt{a_1^2+a_2^2}\sqrt{b_1^2+b_2^2}}$

◉ 좌표 공간에서의 내적

좌표평면을 확장하여 좌표공간에서의 내적을 다음과 같이 정의 한다.

$\vec{a}=(a_1, a_2, a_3)$, $\vec{b}=(b_1, b_2, b_3)$라 할 때,

$\vec{a}\cdot\vec{b}=|\vec{a}||\vec{b}|\cos\Theta$

$=a_1b_1+a_2b_2+a_3b_3$

$\cos\Theta=\dfrac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|}=\dfrac{a_1b_1+a_2b_2+a_3b_3}{\sqrt{a_1^2+a_2^2+a_3^2}\sqrt{b_1^2+b_2^2+b_3^2}}$

벡터 덧셈 뺄셈 소스코드

```
#ifndef __VECTOR_3D_H__
#define __VECTOR_3D_H__

typedef struct vector3d
{
 float x;
 float y;
 float z;

 void (* add)(struct vector3d, struct vector3d, struct vector3d *);
 void (* sub)(struct vector3d, struct vector3d, struct vector3d *);
 void (* scale)(float, struct vector3d);
 void (* dot)(struct vector3d, struct vector3d);
 void (* cross)(struct vector3d, struct vector3d);
} vec3;

void vec3_add(vec3 a, vec3 b, vec3 *r)
{
 r->x = a.x + b.x;
 r->y = a.y + b.y;
 r->z = a.z + b.z;
}

void vec3_sub(vec3 a, vec3 b, vec3 *r)
{
 r->x = a.x - b.x;
 r->y = a.y - b.y;
 r->z = a.z - b.z;
}

#endif
```

그램 슈미츠

```c
#include "vector_3d.h"
#include <stdio.h>

int main(void)
{
 vec3 A = {3, 2, 1};
 vec3 B = {1, 1, 1};
 vec3 X = {1, 0, 0};
 vec3 Y = {0, 1, 0};
 vec3 v[3] = {{0, 4, 0}, {2, 2, 1}, {1, 1, 1}};
 vec3 w[3] = {};
 vec3 R = {0, 0, 0,
     vec3_add, vec3_sub, vec3_scale,
     vec3_dot, vec3_cross, print_vec3,
     gramschmidt_normalization};

 R.add(A, B, &R);
 R.print(R);

 R.sub(A, B, &R);
 R.print(R);

 R.scale(3, R, &R);
 R.print(R);

 printf("A dot B = %f \ n", R.dot(A, B));

 R.cross(X, Y, &R);
 R.print(R);

 R.gramschmidt(v, w, R);

 return 0;
}
```