융합수학 보고서

# R Tilde 행렬과 MATLAB을 이용한 아인슈타인의 특수 상대성이론 시뮬레이션 구현

Simulating Einstein's Special Relativity Using R Tilde Matrix and MATLAB


최 정 담 (崔晸譚 Choi, Jung Dam)

2214

신 주 형 (辛柱衡 Shin, Ju Hyeong)

2308

세종과학예술영재학교



2018

# R Tilde 행렬과 MATLAB을 이용한 아인슈타인의 특수 상대성이론 시뮬레이션 구현

Simulating Einstein's Special Relativity Using R Tilde Matrix and MATLAB

[논문제출 전 체크리스트]

1. 이 논문은 내가 직접 연구하고 작성한 것이다. ☐
2. 인용한 모든 자료(책 논문 인터넷자료 등)의 인용표시를 바르게 하였다. ☐
3. 인용한 자료의 표현이나 내용을 왜곡하지 않았다. ☐
4. 정확한 출처제시 없이 다른 사람의 글이나 아이디어를 가져오지 않았다. ☐
5. 논문 작성 중 도표나 데이터를 조작(위조 혹은 변조)하지 않았다. ☐
6. 다른 친구와 같은 내용의 논문을 제출하지 않았다. ☐

# R Tilde 행렬과 MATLAB을 이용한 아인슈타인의 특수 상대성이론 시뮬레이션 구현

최 정 담

신 주 형

# R Tilde 행렬과 MATLAB을 이용한 아인슈타인의 특수 상대성이론 시뮬레이션 구현


초 록

본 연구에서는 아인슈타인의 특수 상대성 이론, 즉 로렌츠 변환을 해석하며 R Tilde 행렬이라는 개념을 도입하였고, 이 행렬의 다양한 수학적인 성질들을 확인해 보았다. 또 본 연구에서는 R Tilde 행렬을 통한 선형변환을 원, 쌍곡선 등의 다양한 물리적(기하학적인) 상황에 적용하여 그 수학적 의미를 파악해보고자 하였다. 이후에는 행렬 연산에 특화되어 있는 MATLAB이라는 프로그램에 R Tilde 행렬을 활용하여, 특수 상대성이론 시뮬레이션을 제작하였으며, 특수 상대성이론을 학습하는 데 도움이 될 수 있도록 GUI 형태로 이를 변환하였다.



**주제어 (키워드, 색인어)** R Tilde, 특수 상대성이론, 로렌츠 변환, MATLAB, 시뮬레이션

# Contents

# Chapter 1

# 서론

1905년 아인슈타인은 빛의 속도에 가깝게 움직이는 물체들에 대하여 모든 관성계는 동등하고, 모든 관성계에서 진공에서의 빛의 속력은 어디에서나 일정하다는 두 가지 가정을 통하여 특수 상대성이론이라는 새로운 물리 법칙을 제안하였다. 특수 상대성이론의 핵심적인 내용은 모든 좌표계에서 빛의 속력을 일정하게 유지시키기 위해 좌표계의 시간 축과 공간 축이 변화하여야 한다는 내용인데, 이는 빛의 속력에 근접하지 않은 관측자에서는 유의미하게 드러나지 않으므로 일상적인 상황에서는 특수 상대론적인 효과가 잘 드러나지 않고, 그러므로 특수 상대성이론을 직관적으로 이해하는 것은 어려운 편이다.

한편 민코스프키는 민코스프키 기하학이라는 새로운 비유클리드 기하학을 창시하여 아인슈타인의 특수 상대성이론을 수학적으로 해석하고자 하였다. 민코스프키 기하학의 한 가지 특징은 모든 축(시간 축 및 공간 축)이 서로 직교하나 민코스프기 기하 위의 두 좌표 사이의 거리가 식 (1.1) 과 같이 주어진다는 것이다. 식 (1.1)는 $(\Delta t)^2$ 의 부호가 음수라는 것에서 일반적인 유클리드 기하학에서의 두 점 사이의 거리와 차이를 보인다.

$$(\Delta s)^2 = (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2 - (\Delta t)^2 \tag{1.1}$$

민코스프키는 이후 아인슈타인의 특수 상대성이론이 민코스프키 시공간의 좌표 변환이라는 사실을 발견하고, 이 좌표 변환이 두 점 사이의 거리 $(\Delta s)^2$를 변화시키지 않는다는 사실을 알아내었다. 이를 직관적으로 이해할 수 있게 만든 것이 민코스프키 다이어그램, 혹은 민코스프키 시공도식이다. 이는 직관적일 뿐더러, 특수 상대성이론의 물리적 의미가 잘 담겨 있으나, 손으로 민코스프키 시공도식을 그리는 것이 비교적 어렵다는 단점이 있다. 따라서 본 연구는 이러한 민코스프키 시공도식의 강점을 살리면서도, 다양한 물리적 상황에 대하여 쉽게 민코스프키 시공 도식을 작성하고 특수 상대론의 물리적 의미를 이해할 수 있도록 특수 상대성이론 시뮬레이션을 제작하는 데에 주목하였다. 이 과정에서 특수 상대성 이론을 민코스프키 시공간의 좌표축 변환으로 해석하는 민코스프키의 아이디어에 주목하여 시공간 축의 변환을 선형 변환으로 해석하고자 하였고, 이에 R Tilde 행렬을 도입하였다.

# Chapter 2

# 이론적 배경

## 2.1 특수 상대성이론

특수 상대성이론은 1905년 아인슈타인이 발표한 개념으로, 등속도로 움직이는 서로 다른 두 좌표계의 물리량 사이의 관계를 표현하고 있다. 특히 이 관계는 로렌츠 변환이라는 네 개의 등식에 의하여 표현된다. 관성계 $S$에서 본 사건의 좌표를 $(t, x, y, z)$, 관성계 $S'$에서 본 사건의 좌표를 $(t', x', y', z')$라 하면, 서로 다른 두 좌표계에서 본 사건 $E$, $E'$가 $E(0, 0, 0, 0) = E'(0, 0, 0, 0)$를 만족할 때 다음과 같은 식이 성립한다.

$$\begin{aligned} t' &= \gamma(t - \frac{ux}{c^2}) \\ x' &= \gamma(x - ut) \\ y' &= y \\ z' &= z \end{aligned} \tag{2.1}$$

단, 식 (2.1)에서 $\mathbf{u}$는 관성계 $S$에서 본 관성계 $S'$의 속도이며, $u = |\mathbf{u}|$이다. 한편 $\gamma$는 다음과 같이 정의된다.

$$\gamma = \frac{1}{\sqrt{1 - u^2/c^2}}$$

한편 민코프스키 시공간에서, 시공간 좌표는 다음과 같이 벡터로 표현할 수 있다.

$$\vec{E} = \begin{pmatrix} t \\ x \\ y \\ z \end{pmatrix} \tag{2.2}$$

그렇게 했을 때, 식 (2.1)는 다음과 같이 표현할 수 있다.

$$\vec{E'} = L\vec{E}$$
$$\text{where}$$
$$L = \begin{pmatrix} \gamma & -u\gamma & 0 & 0 \\ -u\gamma & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

이것이 민코스프키 시공간을 통해 표현한 로렌츠 변환 행렬식이다.

## 2.2 오일러 회전 변환 행렬

어떤 점 $\vec{P} = \begin{pmatrix} x \\ y \end{pmatrix}$를 반시계방향으로 $\theta$만큼 돌린 점을 $\vec{P} = \begin{pmatrix} x' \\ y' \end{pmatrix}$라고 하자. 이 때 $\vec{P}$와 $\vec{P'}$ 사이에는 다음과 같은 관계가 있다.

$$\vec{P'} = R_\theta \vec{P}$$
$$\text{where}$$
$$R_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{2.4}$$

이 때 $R_\theta$는 다음과 같은 성질을 가지고 있다.

1. 크기 보존; 어떤 도형에 $R_\theta$를 적용한 후의 도형도 원래 도형과 그 크기가 동일하다. 즉,

$$\det(R_\theta) = 1 \tag{2.5}$$

*Proof.*

$$\begin{aligned} &\det(R_\theta) \\ &= \det \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\ &= \cos^2\theta + \sin^2\theta \\ &= 1 \end{aligned}$$

$\square$

2. Homomorphism; $\theta_1+\theta_2$ 만큼 회전시키는 회전 행렬과 $\theta_1$ 과 $\theta_2$ 를 회전시키는 회전 행렬을 각각 수행한 회전 행렬은 같다. 즉,

$$R_{\theta_1}R_{\theta_2} = R_{\theta_1+\theta_2} \tag{2.6}$$

*Proof.*

$$\begin{aligned}
&R_{\theta_1}R_{\theta_2}\\
&= \begin{pmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{pmatrix}\\
&= \begin{pmatrix} \cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2 & \cos\theta_1\sin\theta_2 + \sin\theta_1\cos\theta_2 \\ -\sin\theta_1\cos\theta_2 + \cos\theta_1(-\sin\theta_2) & -\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2 \end{pmatrix}\\
&= \begin{pmatrix} \cos(\theta_1+\theta_2) & \sin(\theta_1+\theta_2) \\ -\sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) \end{pmatrix}\\
&= R_{\theta_1+\theta_2}
\end{aligned}$$

□

한 편 3차원 좌표에서 xy 축으로 회전하는 오일러 회전 행렬은 다음과 같이 표현할 수 있다.

$$R_\theta = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

이를 $n$차 민코스프키 시공도식까지 확장하면 다음과 같다.

$$R_\theta = \underbrace{\begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 & \cdots & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{n \text{ elements}} \tag{2.8}$$

## 2.3 민코스프키 시공도식

민코스프키 시공도식은 특수 상대성이론을 시각화하기 위해 만든 도식으로, 다음과 같은 규칙으로 작성한다.

1. $x$축에는 $\mathbf{u}$의 방향과 같은 축의 위치 축을, $y$축에는 $ct$ 축을 기입한다. 그러므로 빛의 이동 경로는 언제나 $x$축과 $\pm\frac{\pi}{4}$의 각을 이룬다.

2. 로렌츠 변환은 시간 축과 공간 축을 회전시킴으로써 이루어진다. 이 때 회전 각도는 $\tan^{-1}(\frac{v}{c})$이며, 공간 축은 반시계 방향으로, 시간 축은 시계 방향으로 회전한다.

3. 좌표의 스케일 역시 바뀐다. 이 때, 서로 다른 좌표축에서의, 좌표가 $(k, 0)$인 서로 다른 점들의 모임은 $x^2 - t^2 = k^2$을 만족하는 쌍곡선 위에 있다.[1] 한 편 서로 다른 좌표축에서의 좌표가 $(0, l)$인 서로 다른 점들의 모임은 $t^2 - x^2 = l^2$을 만족시킨다.

그림 2.1는 이러한 규칙으로 작성한 민코스프키 시공도식의 예이다.

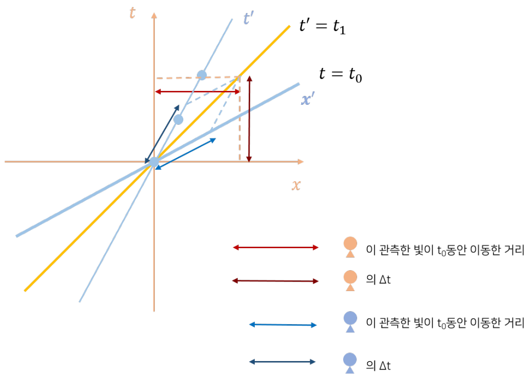


그림 2.1: 민코스프키 시공도식

[1] 식 (1.1)를 참고하여라.

# Chapter 3

# 연구 결과

## 3.1 R Tilde 행렬

### 3.1.1 R Tilde 행렬의 정의와 성질

2차원 상에서의 R Tilde 행렬을 다음과 같이 정의한다.

$$\tilde{R}_\theta = \begin{pmatrix} \cosh\theta & -\sinh\theta \\ -\sinh\theta & \cosh\theta \end{pmatrix} \tag{3.1}$$

이 때 R Tilde 행렬은 오일러 행렬과 다음과 같은 공통점을 가진다.

1. 크기 보존; 어떤 도형에 $\tilde{R}_\theta$를 적용한 후의 도형도 원래 도형과 그 크기가 동일하다. 즉,

$$\det\left(\tilde{R}_\theta\right) = 1 \tag{3.2}$$

*Proof.*

$$\begin{aligned} &\det\left(\tilde{R}_\theta\right) \\ &= \det\begin{pmatrix} \cosh\theta & -\sinh\theta \\ -\sinh\theta & \cosh\theta \end{pmatrix} \\ &= \cosh^2\theta - \sinh^2\theta \\ &= 1 \end{aligned}$$

□

2. Homomorphism; $\theta_1+\theta_2$ 만큼 회전시키는 R tilde 행렬과 $\theta_1$ 과 $\theta_2$ 를 회전시키는 R tilde 행렬을 각각 수행한 회전 행렬은 같다. 즉,

$$\tilde{R}_{\theta_1}\tilde{R}_{\theta_2}=\tilde{R}_{\theta_1+\theta_2} \tag{3.3}$$

*Proof.* $\tilde{R}_\theta$ 의 대각화부터 시작한다.

$$\begin{aligned}\det\left(\tilde{R}-\lambda I\right)&=0\\(\cosh x-\lambda)^2-\sinh^2 x&=0\\(\cosh x-\lambda)&=\pm\sinh x\end{aligned}$$

$$\begin{aligned}\lambda&=\cosh x\pm\sinh x\\&=e^x \quad \text{or} \quad e^{-x}\end{aligned}$$

$$\tilde{R}_\theta=\left(\frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\-1&1\end{pmatrix}\right)\begin{pmatrix}0&e^\theta\\e^{-\theta}&0\end{pmatrix}\left(\frac{1}{\sqrt{2}}\begin{pmatrix}-1&-1\\1&-1\end{pmatrix}\right) \tag{3.4}$$

$$\begin{aligned}\tilde{R}_{\theta_1}\times\tilde{R}_{\theta_2}&=\left(\frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\-1&1\end{pmatrix}\right)\begin{pmatrix}0&e^{\theta_1+\theta_2}\\e^{-(\theta_1+\theta_2)}&0\end{pmatrix}\left(\frac{1}{\sqrt{2}}\begin{pmatrix}-1&-1\\1&-1\end{pmatrix}\right)\\&=\tilde{R}_{(\theta_1+\theta_2)}\end{aligned}$$

□

즉, $\cosh\theta$ 와 $\sinh\theta$ 로 이루어진 $\tilde{R}_\theta$ 는 $R_\theta$ 와 같은 성질을 보인다. 이 성질을 유지하도록 $\tilde{R}_\theta$ 를 $n$ 차까지 확장하면 이는 다음과 같다.

$$\tilde{R}_\theta = \underbrace{\begin{pmatrix} \cosh\theta & -\sinh\theta & 0 & 0 & \cdots & 0 \\ -\sinh\theta & \cosh\theta & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{n\text{ elements}} \tag{3.5}$$

R Tilde 행렬은 $\theta < 0$일 때 x축과 y축을 회전시키되, x축은 반시계방향으로, y축은 시계방향으로 회전시킨다. 이 때 회전 각도는 R Tilde 행렬의 기하적인 의미는 hyperbolic angle의 기하적 의미를 통해 알아낼 수 있다. 원래 x 좌표축에서부터 회전한 x 좌표축 까지 $x^2 - y^2 = 1$의 단위 쌍곡선을 타고 쓸어간 넓이는 정확히 $\frac{\theta}{2}$가 된다. 이는 오일러 회전 행렬에서의 $\theta$의 의미와 공통점이 있는데, 오일러 회전 행렬으로 원래 x 좌표축에서부터 회전한 x 좌표축까지 단위원을 타고 쓸어간 넓이 역시 정확히 $\frac{\theta}{2}$가 되기 때문이다.

### 3.1.2 다양한 곡선의 R Tilde 회전 변환

본 부분부터는 R Tilde 회전 변환에서 $\theta < 0$인 경우를 다룰 것이다. 이 경우에는 $\theta$를 양수로 만들고 R Tilde 행렬을 다음과 같이 적용할 수 있다.

$$\tilde{R}_\theta = \begin{pmatrix} \cosh\theta & \sinh\theta \\ \sinh\theta & \cosh\theta \end{pmatrix}$$

#### 직선

직선의 방정식은 일반적으로 $ax+by+c=0$을 만족한다. 이를 R Tilde 회전 변환시키면

$$\begin{aligned} \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} \cosh\theta & \sinh\theta \\ \sinh\theta & \cosh\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \cosh\theta & -\sinh\theta \\ -\sinh\theta & \cosh\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \\ &= \begin{pmatrix} \cosh\theta x' - \sinh\theta y' \\ -\sinh\theta x' + \cosh\theta y' \end{pmatrix} \end{aligned}$$

$$
\begin{gathered}
ax + by + c = 0 \\
\text{if } \ a \neq \pm b, \text{ wlog,} \\
\sinh\theta_0 x + \cosh\theta_0 y + c = 0 \\
\text{or} \\
\cosh\theta_0 x + \sinh\theta_0 y + c = 0
\end{gathered}
$$

$$
\begin{aligned}
\cosh\theta_0(\cosh\theta x' - \sinh\theta y') + \sinh\theta_0(-\sinh\theta x' + \cosh\theta y') + c = 0 \\
(\cosh\theta_0\cosh\theta - \sinh\theta_0\sinh\theta)x' + (-\cosh\theta_0\sinh\theta + \sinh\theta_0\cosh\theta)y' + c = 0
\end{aligned}
$$

$$
\begin{aligned}
\sinh\theta_0(\cosh\theta x' - \sinh\theta y') + \cosh\theta_0(-\sinh\theta x' + \cosh\theta y') + c = 0 \\
(\sinh\theta_0\cosh\theta - \cosh\theta_0\sinh\theta)x' + (-\sinh\theta_0\sinh\theta + \cosh\theta_0\cosh\theta)y' + c = 0
\end{aligned}
$$

즉 직선을 R Tilde 회전시키면 직선의 hyperbolic angle이 $\theta$만큼 줄어들게 된다. 이는 오일러 회전 행렬과 동일한 결과이다.

## 원

$x^2 + y^2 = 1$의 단위원을 생각하자. 이 원에 R Tilde 행렬을 적용하면 이는 $\frac{\pi}{4}$만큼 회전한 타원이 나온다.[1]

$$
\begin{aligned}
\begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \cosh\theta & \sinh\theta \\ \sinh\theta & \cosh\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\
\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \cosh\theta & -\sinh\theta \\ -\sinh\theta & \cosh\theta \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \\
&= \begin{pmatrix} x'\left(\frac{1}{\sqrt{2}}\cosh\theta + \frac{1}{\sqrt{2}}\sinh\theta\right) + y'\left(\frac{1}{\sqrt{2}}\cosh\theta - \frac{1}{\sqrt{2}}\sinh\theta\right) \\ y'\left(\frac{1}{\sqrt{2}}\cosh\theta - \frac{1}{\sqrt{2}}\sinh\theta\right) + x'\left(\frac{1}{\sqrt{2}}\cosh\theta + \frac{1}{\sqrt{2}}\sinh\theta\right) \end{pmatrix}
\end{aligned}
$$

$$
x^2 + y^2 = 1
$$

[1] $-\frac{\pi}{4}$ 만큼 회전했다는 것과 논리적으로 동치이다.

$$\left\{x'\left(\frac{1}{\sqrt{2}}\cosh\theta+\frac{1}{\sqrt{2}}\sinh\theta\right)+y'\left(\frac{1}{\sqrt{2}}\cosh\theta-\frac{1}{\sqrt{2}}\sinh\theta\right)\right\}^2$$
$$+\left\{y'\left(\frac{1}{\sqrt{2}}\cosh\theta-\frac{1}{\sqrt{2}}\sinh\theta\right)+x'\left(\frac{1}{\sqrt{2}}\cosh\theta+\frac{1}{\sqrt{2}}\sinh\theta\right)\right\}^2=1$$

$$x'^2(\cosh\theta+\sinh\theta)^2+y'^2(\cosh\theta-\sinh\theta)^2=1$$

이 결과는 Geogebra로도 확인할 수 있었다.

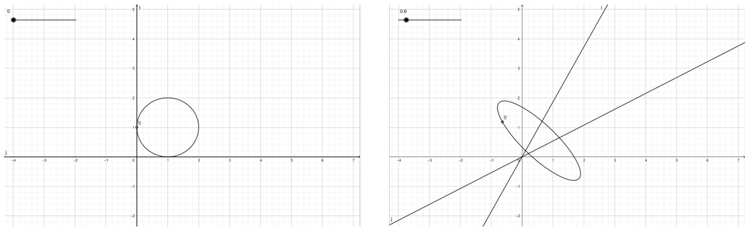

그림 3.1: 원의 R Tilde 회전, 우측 타원은 회전한 좌표축에서 본 원의 모습

**쌍곡선**

단위 쌍곡선을 R Tilde 회전 변환하면 단위 쌍곡선이 나온다. 이는 단위 원을 오일러 회전 변환하면 단위 원이 된다는 사실과 연결된다.

$$x^2-y^2=1$$

$$(\cosh\theta x'-\sinh\theta y')^2-(-\sinh\theta x'+\cosh\theta y')^2=1$$
$$(\cosh^2\theta-\sinh^2\theta)x'^2+(\sinh^2\theta-\cosh^2\theta)y'^2=1$$

$$x'^2-y'^2=1$$

## 3.2 MATLAB 시뮬레이션

### 3.2.1 로렌츠 변환의 행렬 표현

로렌츠 변환은 원점이 같고, $x$방향으로만 이동하는 관성계에 대해 식 (2.3)으로 표현된다. 이를 R Tilde 행렬을 이용하여 표현하면 다음과 같다.

$$
\begin{gathered}
L = \tilde{R}_\theta \\
\text{where} \\
\theta = \gamma
\end{gathered}
\tag{3.6}
$$

한편 관성계의 원점을 맞추고, 관성계가 $x$축으로 움직이도록 하게 만들면 최종 식은 다음과 같아진다.

$$O_i' = R_\theta \times \tilde{R}_{\theta'} \times R \times (O_i - S) \tag{3.7}$$

여기에서 각 행렬은 다음과 같다.

$$S_{ij} = O_{i1}$$

$$R_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \text{ where } \theta = \tan^{-1}\frac{v_y}{v_x}$$

$$\tilde{R}_{\theta'} = \begin{pmatrix} \cosh\theta' & -\sinh\theta' & 0 \\ -\sinh\theta' & \cosh\theta' & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ where } \theta' = \cosh^{-1}\gamma$$

### 3.2.2 Text 기반 프로그램 작성 코드

최종 목표를 달성하기 이전에 Text 기반으로 입출력을 받아 특수 상대성 이론을 시뮬레이션하는 프로그램을 제작하였다. 다양한 물체의 초기 위치, 속도, 아주 작은 크기의 가속도를 입력하면, 해당 물체들 중 하나를 기준으로 민코스프키 시공도식과 실제 물체 움직임을 애니메이션으로 출력하게 된다. 해당 프로그램의 실행 모습은 그림 3.2, 2.1, 3.4과 같다.

```
clc ;
clear all ;
disp('SoPhiSTIcaTeD AnD DiMeN presents, SR simulation. <Info
    : c=1>') ;
tmax = input('What is max t? ') ;
objnum = input('What is object number? ') ;
t=0:0.01:tmax;

objecti = cat(1,t,zeros(2,size(t,2))) ;
object = objecti ;
```

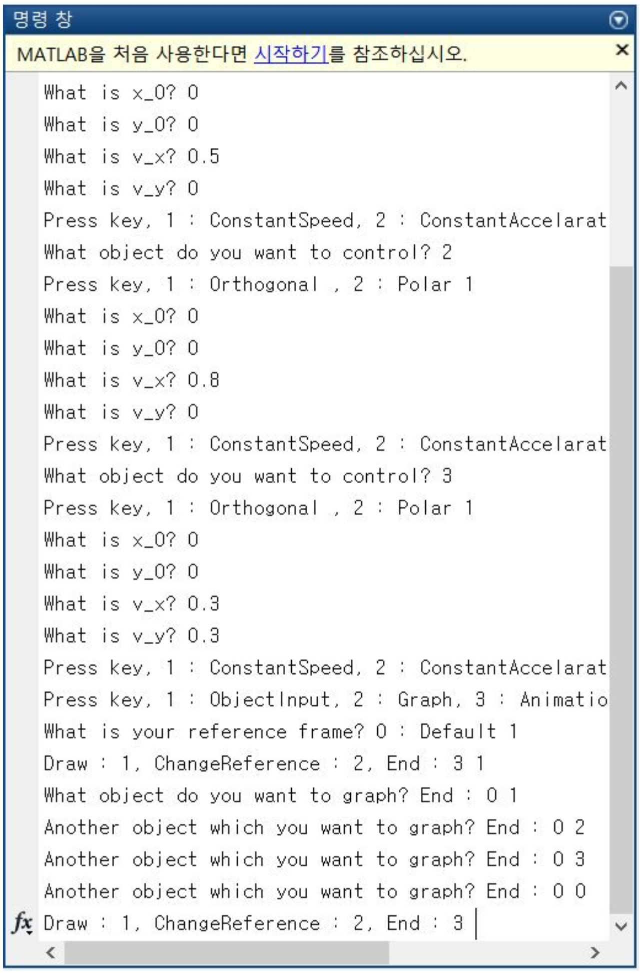


그림 3.2: Text 창

```
objctrl = 0;

for i=1:objnum-1
    object = cat(3,object,objecti) ;
end


isobjref = zeros(1,objnum);
objref = 0 ;
velocity = zeros(2,objnum);
accelaration = zeros(1,objnum);
```

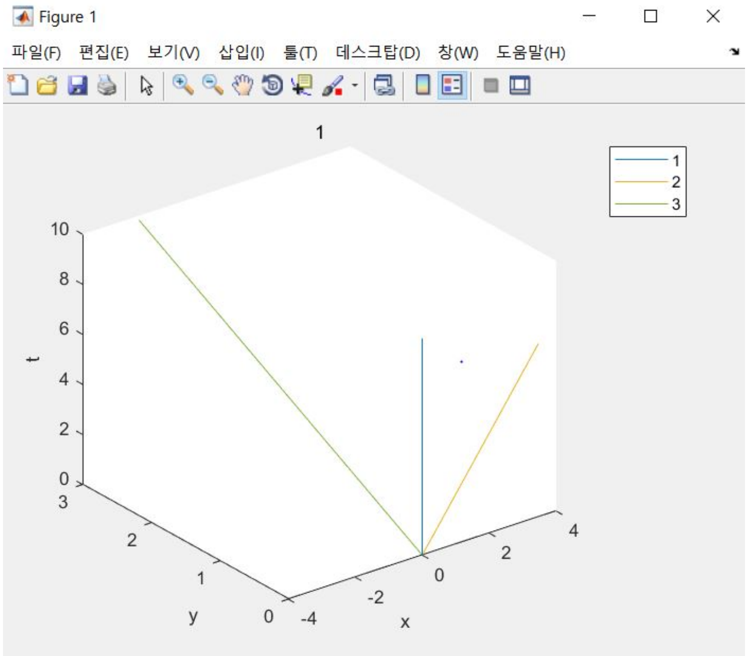


그림 3.3: 민코스프키 시공도식

```
key=input('Press key, 1 : ObjectInput, 2 : Graph, 3 : Animation, 4 : End ') ;

while(key ~= 4)
        switch key

            case 1

                key1=input('Press key, 1 : ConstantSpeed, 2 : ConstantAccelaration, 3 : Otherwise , 4 : End ') ;
                while(key1 ~= 4)

                switch key1
                    case 1
                        objctrl = input('What object do you want to control? ');
                        while(objctrl==0 || objctrl>objnum)
```

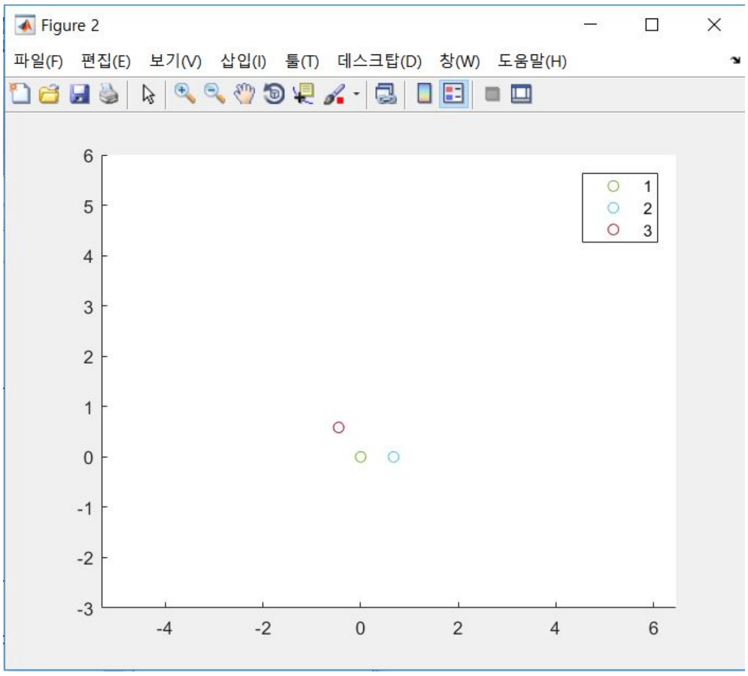


그림 3.4: 애니메이션

```
        objctrl = ('Invalid Object#. What object do you want to control? ');
    end
    key11 = input('Press key, 1 : Orthogonal , 2 : Polar ') ;
    isobjref(objctrl) = 1 ;
    switch key11
        case 1
            x_0 = input('What is x_0? ');
            y_0 = input('What is y_0? ');
            v_x = input('What is v_x? ');
            v_y = input('What is v_y? ');
            velocity(1,objctrl) = sqrt(v_x^2+v_y^2) ;
```

```
                velocity(2,objctrl) = atan(v_y/v_x) ;
                object(2,:,objctrl) = x_0 + object(1,:,objctrl) .* v_x ;
                object(3,:,objctrl) = y_0 + object(1,:,objctrl) .* v_y ;

            case 2
                r = input('What is r? ');
                r_theta = input('What is theta? ');
                x_0 = r * cos(r_theta) ;
                y_0 = r * sin(r_theta) ;
                velocity(1,objctrl) = input('What is v? ') ;
                velocity(2,objctrl) = input('What is theta? ') ;
                v_x = velocity(1,objctrl) * cos(velocity(2,objctrl)) ;
                v_y = velocity(1,objctrl) * sin(velocity(2,objctrl)) ;
                object(2,:,objctrl) = x_0 + object(1,:,objctrl) .* v_x ;
                object(3,:,objctrl) = y_0 + object(1,:,objctrl) .* v_y ;

            otherwise
        end

    case 2
        objctrl = input('What object do you want to control? ');
        while(objctrl==0 || objctrl>objnum)
```

```
            objctrl = ('Invalid Object#. What object do you want to control? ');
        end
        key21 = input('Press key, 1 : Orthogonal , 2 : Polar ') ;
        switch key21
            case 1
                x_0 = input('What is x_0? ');
                y_0 = input('What is y_0? ');
                v_x = input('What is v_x0? ');
                v_y = input('What is v_y0? ');
                a_x = input('What is a_x? ');
                a_y = input('What is a_y? ');

                velocity(1,objctrl) = sqrt(v_x^2+v_y^2) ;
                velocity(2,objctrl) = atan(v_y/v_x) ;
                object(2,:,objctrl) = x_0 + object(1,:,objctrl) .* v_x + 0.5 .* object(1,:,objctrl).^2 .* a_x ;
                object(3,:,objctrl) = y_0 + object(1,:,objctrl) .* v_y + 0.5 .* object(1,:,objctrl).^2 .* a_y ;


            case 2
                r = input('What is r? ');
```

```
                        r_theta = input('What is theta? ');
                        x_0 = r * cos(r_theta) ;
                        y_0 = r * sin(r_theta) ;
                        velocity(1,objctrl) = input('What is v? ') ;
                        velocity(2,objctrl) = input('What is theta? ') ;
                        a = input('What is a? ') ;
                        a_theta = input('What is theta? ') ;
                        v_x = velocity(1,objctrl) * cos(velocity(2,objctrl)) ;
                        v_y = velocity(1,objctrl) * sin(velocity(2,objctrl)) ;
                        a_x = a * cos(a_theta) ;
                        a_y = a * sin(a_theta) ;
                        object(2,:,objctrl) = x_0 + object(1,:,objctrl) .* v_x + 0.5 .* object(1,:,objctrl).^2 .* a_x ;
                        object(3,:,objctrl) = y_0 + object(1,:,objctrl) .* v_y + 0.5 .* object(1,:,objctrl).^2 .* a_y ;




                    otherwise
                end

            case 3
                disp('No. ') ;
            otherwise
        end
```

```
            key1=input('Press key, 1 : ConstantSpeed, 2 : ConstantAccelaration, 3 : Otherwise , 4 : End ') ;
            end
        case 2
            objref = input('What is your reference frame ? 0 : Default ');
            while(objref~=0 && isobjref(max(1,objref))~=1)
                objref = input('That is not a reference frame. Input again. 0 : Default ');
            end

            if(objref~=0)
                        theta_ref = velocity(2,objref) ;
                        v_ref = velocity(1,objref) ;
                        theta_ref_tilde = acosh(1/sqrt(1-v_ref.^2)) ;
                        x_0 = object(2,1,objref);
                        y_0 = object(3,1,objref);
                        R = [1 0 0 ; 0 cos(-theta_ref) -sin(-theta_ref) ; 0 sin(-theta_ref) cos(-theta_ref)] ;
                        R_tilde = [ cosh(theta_ref_tilde) sinh(theta_ref_tilde) 0 ; sinh(theta_ref_tilde) cosh(theta_ref_tilde) 0 ; 0 0 1 ] ;
                        S = cat(1, zeros(1,size(t,2)) , -x_0*ones(1,size(t,2)) , -y_0*ones(1,size(t,2))) ;
                        objectg = zeros(size(object));
                        for i = 1:objnum
                        objectg(:,:,i) = eye(3)/R/R_tilde*R*(object(:,:,i)-S) ;
                        end

```

```
                else
                    objectg = object ;
        end

        key21 = input('Draw : 1, ChangeReference : 2, End : 3 ') ;

        while( key21 ~= 3 )
            switch key21

                case 1


                    objctrl=input('What object do you want to graph? End : 0 ');
                    if(objctrl~=0)

                    hplot = plot3(objectg(2,:,objctrl), objectg(3,:,objctrl), objectg(1,:,objctrl), 'DisplayName', int2str(objctrl)) ;
                    hold all;

                    objctrl=input('Another object which you want to graph? End : 0 ');
                    while(objctrl~=0)
                    plot3(objectg(2,:,objctrl), objectg(3,:,objctrl), objectg(1,:,objctrl), 'DisplayName', int2str(objctrl)) ;
                    hplot = cat(1, hplot, plot3(objectg(2,:,objctrl), objectg(3,:,objctrl), objectg(1,:,objctrl), 'DisplayName',
```

```
            int2str(objctrl)) );
        objctrl=input('Another object which you want to graph? End: 0 ');
        end


        xlabel('x')
        ylabel('y')
        zlabel('t')
        legend(hplot)
        title(int2str(objref))
        hold all ;

        end
    case 2
        objref = input('What is your reference frame? 0 : Default ');
        while(objref~=0 && isobjref(max(1,objref))~=1)
            objref = input('That is not a reference frame. Input again. 0 : Default ');
        end

        if(objref~=0)
        theta_ref = velocity(2,objref) ;
        v_ref = velocity(1,objref) ;
        theta_ref_tilde = acosh(1/sqrt(1-v_ref.^2)) ;
        R = [1 0 0 ; 0 cos(-theta_ref) -sin(-theta_ref) ; 0 sin(-theta_ref) cos(-theta_ref)] ;
        R_tilde = [ cosh(theta_ref_tilde) sinh(theta_ref_tilde) 0 ; sinh(theta_ref_tilde) cosh(
```

```
                        theta_ref_tilde) 0 ; 0 0 1 ];
                        S = cat(1, zeros(1,size(t,2)) , -x_0*ones(1,size(t,2)) , -y_0*ones(1,size(t,2))) ;
                        objectg = zeros(size(object));
                        for i = 1:objnum
                        objectg(:,:,i) = eye(3)/R/R_tilde*R*(object(:,:,i)-S) ;
                        end

                        else
                            objectg = object ;
                        end

                    otherwise
                        ('Work Dimen!!!!') ;
                end
                key21 = input('Draw : 1, ChangeReference : 2, End : 3 ') ;
            end

        case 3
            objref = input('What is your reference frame? 0 : Default ');
                        while(objref~=0 && isobjref(max(1,objref))~=1)
                            objref = input('That is not a reference frame. Input again. 0 : Default ');
                        end

                        if(objref~=0)
                        theta_ref = velocity(2,objref) ;
                        v_ref = velocity(1,objref) ;
                        theta_ref_tilde = acosh(1/sqrt(1-v_ref.^2)) ;
```

```
            x_0 = object(2,1,objref);
            y_0 = object(3,1,objref);
            R = [1 0 0 ; 0 cos(-theta_ref) -sin(-theta_ref) ; 0 sin(-theta_ref) cos(-theta_ref)] ;
            R_tilde = [ cosh(theta_ref_tilde) sinh(theta_ref_tilde) 0 ; sinh(theta_ref_tilde) cosh(theta_ref_tilde) 0 ; 0 0 1 ] ;
            S = cat(1, zeros(1,size(t,2)) , -x_0*ones(1,size(t,2)) , -y_0*ones(1,size(t,2))) ;
            objectg = zeros(size(object));
            for i = 1:objnum
            objectg(:,:,i) = eye(3)/R/R_tilde*R*(object(:,:,i)-S) ;
            end

            else
                objectg = object ;
            end

    objani = 0;
    objani_scan = input('What object do you want to animate? End : 0') ;

    while(objani_scan~=0)
        objani = cat(2,objani,objani_scan) ;
        objani_scan = input('Extra object to animate? End : 0') ;
    end

    if(max(objani)~=0)
        objani = objani(2:size(objani,2)) ;
        labels = string(objani) ;
        objecta = objectg(:,:,objani) ;
```

```
            figure
            xlim([min(min(objecta(2,:,:)))-3,max(max(
                objecta(2,:,:)))+3])
            ylim([min(min(objecta(3,:,:)))-3,max(max(
                objecta(3,:,:)))+3])
            hold on;
            for k = 1:size(t,2)
                if k > 1
                    delete(hpoints);
                end

                    for p = 1:size(objani,2)
                    hpoints(p) = plot(objecta(2,k,p),
                        objecta(3,k,p),'o');
                    end
                    legend(labels);
                    pause(0.001)
                    drawnow limitrate
            end

            end

        otherwise


    end
    key=input('Press key, 1 : ObjectInput, 2 : Graph, 3 :
        Animation, 4 : End ') ;
end
```

### 3.2.3 GUI 프로그램

Text 기반 시뮬레이션 프로그램을 참조하여, MATLAB의 GUI 제작 프로그램을 활용하여 GUI 프로그램을 제작하였다. 해당 프로그램의 작동 모습은 다음과 같다.

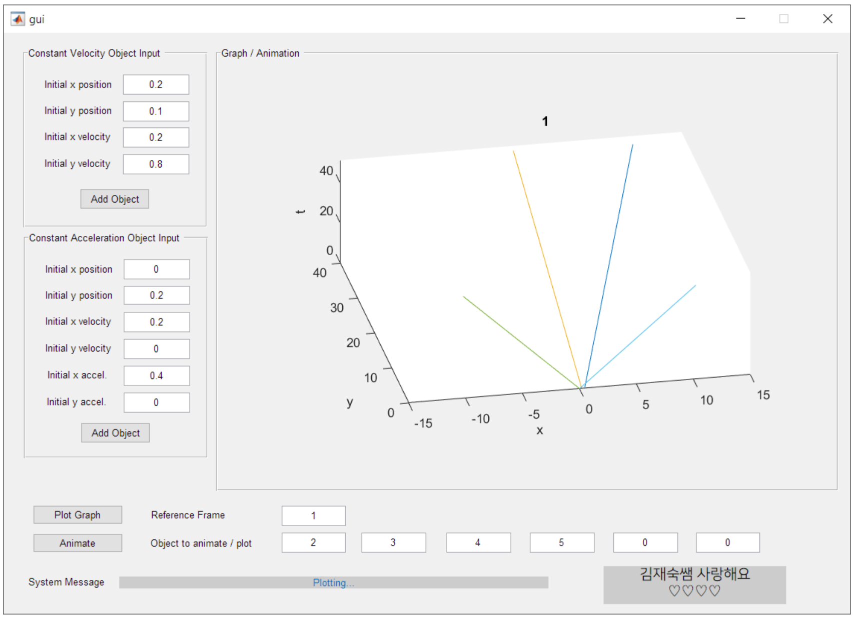


그림 3.5: GUI 프로그램

# Chapter 4

# 결론

본 연구에서는 특수상대성이론을 선형변환으로 해석하여, R Tilde 행렬이라는 쌍곡삼각함수로 이루어진 회전 행렬에 대하여 탐구해 보았다. 그 결과 R Tilde 행렬은 오일러 회전 행렬과 여러 가지 공통점을 가지고 있었으며, 이는 다양한 이차곡선의 변환에서도 드러났다.

한편 본 연구에서는 MATLAB을 이용하여 특수상대성이론에서 일어나는 많은 현상을 시뮬레이션 할 수 있는 프로그램을 제작하였다. 이 과정에서 R Tilde 행렬을 이용하여 일반적인 경우에 적용할 수 있는 로렌츠 변환 행렬을 만들어내었다. 제작 결과 등속도 혹은 약간의 가속도를 가진 물체들이 관성계에서 어떻게 보이는 지 민코프스키 시공도식을 그리고 애니메이션을 제작하였다.

본 연구의 가치는 직관적으로 받아들이기 어려운 특수 상대성이론에 대해 다양한 경우를 테스트할 수 있는 시뮬레이션 프로그램을 개발하였다는 데에 그 의의가 있다. 또한 이 과정에서 MATLAB에게 특화되어 있는 행렬 연산을 위해 로렌츠 변환 행렬을 일반적인 경우에 대하여 제작하였으며, 이 과정에서 R Tilde 행렬의 다양한 수학적인 성질을 알아보았다는 의의도 있다.

# Bibliography


[1] Schutz, A First Course in General Relativity

[2] Novice Math and Science:

[3] Minkowski Spacetime Diagram Wikipedia

[4] Khan Academy: Lecture on Minkowski Spacetime Diagram and Derivations of Lorentz Transformation

[5] S.Friedberg, A.Insel, L.Spence, Linear Algebra