

卒業論文 2014 年度（平成 26 年度）

# 降臨鉄道：模型モノレールを利用した遠隔通信

慶應義塾大学 環境情報学部 環境情報学科

山田 尚昭

増井俊之 研究室

2015 年 1 月

## 降臨鉄道: 模型モノレールを利用した遠隔通信

### 論文要旨

カメラを登載した模型モノレールをオフィスの天井で走らせることによって、どこからでもオフィスの様子を覗いたりオフィス内の人間と会話したりできる遠隔通信システムを作成した。車輪で移動可能なロボットを利用することによって遠隔地のユーザが会議や学会に参加する試みが近年盛んになっているが、混雑した環境ではロボットが自由に移動できないため実運用が難しいことが多い。邪魔物が無い天井に装着したレール上を自由に移動できるモノレールを利用することにより、実用的な遠隔コミュニケーションシステムが実現できた。

### キーワード

遠隔コミュニケーション, テレプレゼンス, ユーザインタフェース

慶應義塾大学 環境情報学部 環境情報学科

山田 尚昭

## **Camera on Rails: Telecommunication with Model Monorail Trains**

### **Summary**

We developed the “Camera on Rails” telecommunication system with which a user can communicate with other people in a distance office through a camera on a model monorail train running on the ceiling of the office. Using our system, the user can monitor the current status of the office by running the train on the ceiling and have conversation with the people in the office.

### **Keywords**

Telecommunication, Telepresence, Userinterface

Faculty of Environment and Information Studies  
Keio University

Naoaki Yamada

# 目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	1
第2章	テレプレゼンス	2
2.1	O B降臨システム	2
2.2	降臨鉄道	3
2.3	関連研究	3
第3章	実装	4
3.1	システム構成	4
3.2	降臨鉄道	4
3.2.1	懸垂式モノレール型プラレール	4
3.2.2	Android	5
3.3	Webアプリケーション	5
3.3.1	ユーザ用アプリケーション	5
3.3.2	Android用アプリケーション	5
3.4	linda-server	6
3.4.1	Linda	6
3.4.2	降臨鉄道での実装内容	6
第4章	評価	9
第5章	おわりに	10
	謝辞	11

参考文献	12
付録A    Webアプリケーションプログラム	13
android.js    . . . . .	13
pc.js    . . . . .	14
付録B    モータ制御回路	15
回路図    . . . . .	15

# 第1章 はじめに

## 1.1 研究背景

遠隔地の人と同じ場所にいるかのような感覚を強化するテレプレゼンスシステムの研究が盛んであり、近年では、自宅警備、家族との会話、ビデオ会議などを目的としたテレプレゼンスロボットが販売されている。

2014年に開催されたWISS2014では、Double RoboticsのDouble[1]というテレプレゼンスロボットの導入により、遠隔地の人間が会議に参加可能になっていた(図 1.1)。

## 1.2 研究目的

筆者の研究室では、卓上を動き回ることのできるロボットを利用した遠隔コミュニケーション支援システム「OB降臨システム(図 1.2) [2]」を作成して運用している。「OB降臨システム」は研究室内にいる学生と普段面識のない研究室OBの円滑な遠隔コミュニケーション支援を行うシステムである。

ケーブルの制約のため動ける範囲が狭く、机の上の障害物のために移動が制限されたりカメラからの映像が見られなかったりすることが多かった。この問題はOB降臨システムだけではなく、同系統の卓上ロボットや人を模したテレプレゼンスロボットにも言えることだろう。

邪魔物が存在しない天井を移動する遠隔情報共有システム「降臨鉄道」を実装した。



図 1.1 Doubleを利用した質疑応答

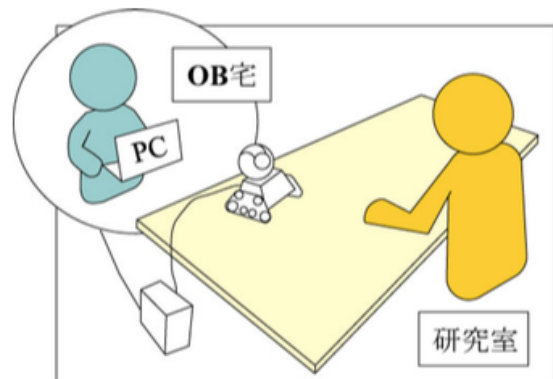


図 1.2 OB降臨システムイメージ図

## 第2章 テレプレゼンス

テレプレゼンスとは遠隔地の人間と実際に対面しているかのような臨場感を提供する技術のことである。テレプレゼンスの研究には2通りのアプローチがある。1つはその場にいるかのような臨場感を提供することで遠隔での作業を実現するもの。もう1つは遠隔地の相手と実際に対面している感覚を強化してより自然なコミュニケーションを取ることを目指すものである。前者は災害時に使われるレスキューロボットのように危険な場所などでの作業に適している。後者はテレビ電話やビデオ会議など、作業ではなくコミュニケーションを目的とした場面で活躍している。

### 2.1 OB降臨システム

OB降臨システムは筆者の研究室で運用されているガンタンク（「機動戦士ガンダム」に登場する有人式人型ロボット兵器）のプラモデルを介した遠隔コミュニケーション支援システムである。2ちゃんねる等に有名人が書き込みをすることを「降臨」と呼ぶことになぞらえて、ロボットに乗り移って研究室に「降臨」したOBとの円滑な会話を実現する。

テレプレゼンスを強化するあまり、親しくない人と話たりOBや有名人など畏怖すべき存在が目の前にいたりするという心理的障壁をOB降臨システムでは問題として掲げている。その解決策として、あえて小型のロボットに降臨させることでOBに小物感を出しテレプレゼンスを低下させることで両者の円滑なコミュニケーションを支援した。

しかし、OB降臨システムはケーブルの制約のため動ける範囲が狭く、机の上の障害物のために移動が制限されたりカメラからの映像が見られなかったりすることが多かった（図 2.1）。



図 2.1 障害物に阻まれるガンタンク



図 2.2 降臨鉄道

## 2.2 降臨鉄道

降臨鉄道（図 2.2）はカメラを搭載した模型モノレールを利用した円滑なコミュニケーション支援システムである。ユーザはWebページにアクセスすることによって遠隔地のリアルタイム映像を見ながら天井にあるモノレールを移動させたり、遠隔地のユーザと会話したりすることができる。混雑した環境ではロボットが自由に移動できないため実運用が難しいことが多いが、邪魔物が無い天井に装着したレール上を自由に移動できるモノレールを利用することにより、実用的な遠隔コミュニケーションシステムが実現できた。

## 2.3 関連研究

明興産業は工場の天井にレールを設置して空間監視管理を行うロボットを製造している[3]。ロボットには各種のセンサが積まれており、工場内での機械・プラントラインの操業状況や動作状態の監視や管理、異臭や異音の検知といった用途に利用されている。

Squarenetは天井を「唯一の空き空間」とし空間移動ロボットシステムを製造している[4]。天井空間を使用することで既存の設備レイアウトを変更することなく、工場内全域のロボット化を実現することができる。



## 第3章 実装

### 3.1 システム構成

降臨鉄道システムは，Androidスマートフォンが取り付けられた懸垂式モノレール型のプラレール（タカラトミーの鉄道玩具）とWebページ，linda-serverから構成される（図 3.1）．

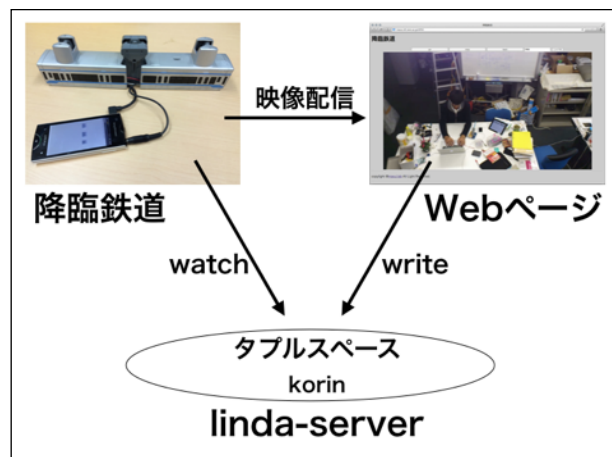


図 3.1 降臨鉄道システム構成図

### 3.2 降臨鉄道

#### 3.2.1 懸垂式モノレール型プラレール

懸垂式モノレール型プラレールの筐体内部にはAndroidスマートフォンから出力された440Hz正弦波信号に基づいてリレーを駆動させることによってDCモータでの前進/後退を制御する回路が実装されている．Androidスマートフォンから出力された正弦波信号をオペアンプで増幅し整流平滑したものでトランジスタを駆動させリレーを駆動している．

電源はUSB接続のモバイルバッテリーを使用し，オペアンプやリレー，DCモータへの電源，Androidスマートフォンの給電を行っている．

### 3.2.2 Android

使用したAndroidスマートフォンはSonyのXperia rayである。

懸垂式モノレール型プラレールにはAndroidスマートフォンが取り付けられている。このAndroidスマートフォンのイヤホン端子は懸垂式モノレール型プラレール筐体内部のDCモータを制御する回路に繋がっている。移動指示に従って正弦波信号をL/Rイヤホン端子に出力する。

カメラからの映像の配信にはAndroidアプリケーションIP Webcam[5]を使用し、MotionJPEG形式の映像をHTTP通信で配信している。

## 3.3 Webアプリケーション

### 3.3.1 ユーザ用アプリケーション

降臨鉄道のユーザはブラウザからWebページにアクセスすることで降臨鉄道を利用できる。Webページを開くと研究室のリアルタイム映像と降臨鉄道の操作インタフェースが表示される（図 3.3）。

研究室のリアルタイム映像は、Androidスマートフォンから配信されているMotionJPEG形式の映像をHTMLのIMGタグで埋め込むことで表示している。

降臨鉄道の操作は左に進む、右に進む、停止するの3つのボタンを押すことでlinda-serverに移動指示を送っている。

### 3.3.2 Android用アプリケーション

懸垂式モノレール型プラレールに取り付けられたAndroidスマートフォンはブラウザからAndroidスマートフォン用ページにアクセスすることで、ユーザ用ページからの移動指示に従ってモータを制御するプログラムを実行することができる。

Androidスマートフォンは移動指示を受け取ると、440Hz正弦波信号をイヤホン端子から出力、つまり、440Hzの音のMP3ファイルを再生する。停止指示があった場合はMP3ファイルの再生を停止する。

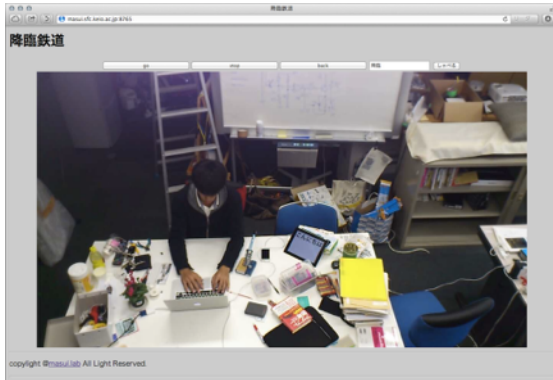


図 3.2 ユーザ用のWebページ

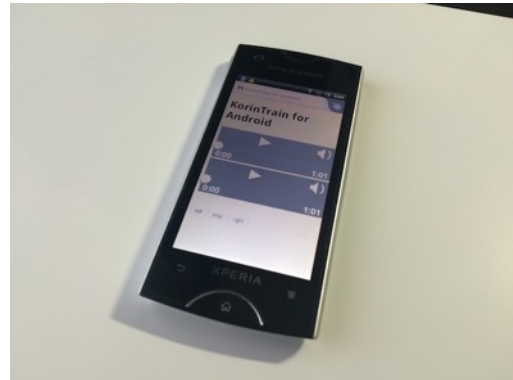


図 3.3 AndroidのWebページ

## 3.4 linda-server

操作は並列計算プリミティブLindaをWebサーバ上に実装したlinda-server[6]を用いて実装している。クライアントはJavaScriptで実装した。

### 3.4.1 Linda

Lindaとは1980年代にイエール大学で生まれた並列処理を行うための実装モデルである。タプルスペース(tuplespace)と呼ばれる共有メモリ空間に、型付きのデータレコード(タプル)を格納する。in/out/rd/inp/rdp という命令で操作することで、大抵の並列処理が記述できる。

今回は、博士課程の橋本翔氏が開発したlinda-serverというサーバ環境を用いて実装を行った。linda-serverは研究室で運用しているもの (<http://linda-server.herokuapp.com>) を利用した。

### 3.4.2 降臨鉄道での実装内容

ユーザが利用するWebページから linda-server に書き込むため、タプルを送信する側のサーバを実装する。

```
var server_url = "http://linda-server.herokuapp.com";
var socket = io.connect(server_url);
var linda = new Linda().connect(socket);
var ts = linda.tuplespace("korin");
linda.io.on("connect", function(){});
```

linda-server に korin というタプルスペースを作成し接続する。

これで linda-server.herokuapp.com/korin との通信が可能になった。

次に、右に進む、左に進む、停止するボタンを押した時にタプルを korin に送信する。

```
function move_right(){
ts.write({type: "move", cmd: "right"});
}
function move_left(){
ts.write({type: "move", cmd: "left"});
}
function move_stop(){
ts.write({type: "move", cmd: "stop"});
}
```

この write によって korin にタプルが書き込まれる。

この状態を linda-server.herokuapp.com/korin で確認すると、以下のように linda-server の korin にタプルが書き込まれていることが分かる (図 3.4)。



図 3.4 linda-serverのkorinタプルスペース

次に、Androidスマートフォンから linda-server の命令を読み取るクライアント側を実装する。

先程の送信側サーバと同じく korinに 接続し、タプルが書き込まれたら watch して読み取る。これによりリアルタイムな送受信が可能となる。

```
    ts.watch({type: "move", cmd: "left"}, function(err, tuple){
play_left();
    });
    ts.watch({type: "move", cmd: "right"}, function(err, tuple){
play_right();
    });
    ts.watch({type: "move", cmd: "stop"}, function(err, tuple){
play_stop();
    });
```

## 第4章 評価

天井にカメラを設置するというと監視カメラや防犯カメラを思い浮かべる人が多いだろう。確かに実際に運用では研究室の様子をくまなく見ることができた。OB降臨システムの場合は見られたくないときに卓上のロボットを隠してしまうということが可能だったが、降臨鉄道は天井にあるためそれが不可能である。研究室はパブリックな空間であり見られて困るようなことはないのだが、自分以外は誰もいない部屋なのに見られているかもしれないという不安は付き纏う。研究室にいる人の「見られてもよい」と降臨する人の「見たい」が同時にあるときにだけ動作できる仕組みがあると良いと考える。

また、今後の課題として次の点が挙げられる。

### (1)複数の人が参加できない

本実装では使用している模型モノレールが1つしかないため、複数の人が同時にアクセスした場合1つの模型モノレールを共同で使うことになる。模型モノレールの操作が重複した場合、ユーザは思った操作ができないと考えられる。誰かが操作している間にアクセスした場合は後からアクセスした人は模型モノレールの操作ができないようにするなど複数人が同時にシステムを利用することができる仕組みが必要である。

### (2)任意の場所への移動

レールに沿って走るという制約があるため、レールのない場所に移動したり任意の場所に瞬時に移動したりすることができない。カメラの角度を変えたりズームできるようにしたりするなどの工夫が必要である。

### (3)「駅」での充電

本実装では電源に使用しているモバイルバッテリーの充電が切れた時に、天井に登って人間の手で充電済みのバッテリーと交換をしている。バッテリーの充電を自動で行う「駅」を設置することで運用する上での負担を減らすことができるだろう。

## 第5章 おわりに

本研究では，カメラを登載した模型モノレールをオフィスの天井で走らせることによって，どこからでもオフィスの様子を覗いたりオフィス内の人間と会話したりできる遠隔通信システム「降臨鉄道」作成した．今回のデモで紹介したシステムは現在，研究室内で実際の運用を通じて実験中である．

降臨鉄道は遠隔コミュニケーションシステムとして利用するのはもちろんのこと，多くの人が集まるイベント会場などで遠隔イベント参加システムとして利用できるだろう．

将来的には目的に応じたロボットを複数個用意し，世代間を超えてより活発な研究活動ができるようにしたいと考えている．

## 謝辞

本研究を進める上で、多大なご指導を下さった増井俊之教授，実装に協力いただいた橋本翔氏，中山拓哉氏に大変感謝を申し上げます。

また，研究を進める上で様々なアドバイスをくださった増井研究室の皆様にも御礼申し上げます。



## 参考文献

- [1] Double Robotics - Telepresence Robot for Telecommuters, <http://www.doublerobotics.com>
- [2] 廣田馨, 橋本翔, 増井俊之 : OB降臨システム, インタラクション2011, (2011).
- [3] 明興産業株式会社, <http://www.meikos.co.jp/robo.html>
- [4] 天井空間移動ロボットシステム, <http://www.squarenet.co.jp/squaremeister-j.htm>
- [5] IP Webcam - Google Play, <https://play.google.com/store/apps/details?id=com.pas.webcam>
- [6] node-linda/linda GitHub, <https://github.com/node-linda/linda>

## 付録A Webアプリケーションプログラム

### android.js

```
var server_url = "http://linda-server.herokuapp.com";
var socket = io.connect(server_url);
var linda = new Linda().connect(socket);
var ts = linda.tuplespace("korin");

linda.io.on("connect", function({}));
ts.watch({type: "move", cmd: "left"}, function(err, tuple){
  if (right.paused == true && left.paused == true) {
    play_left();
  } else if (right.paused == false && left.paused == true){
    play_stop();
    setTimeout("play_left()", 500);
  }
});

ts.watch({type: "move", cmd: "right"}, function(err, tuple){
  if (right.paused == true && left.paused == true) {
    play_right();
  } else if (right.paused == true && left.paused == false){
    play_stop();
    setTimeout("play_right()", 500);
  }
});

ts.watch({type: "move", cmd: "stop"}, function(err, tuple){
  play_stop();
});

function play_left() {
  left.play();
}

function play_right() {
  right.play();
}

function play_stop() {
  if (right.paused == false) {
    right.pause();
    right.currentTime = 0;
  }
  if (left.paused == false) {
    left.pause();
    left.currentTime = 0;
  }
}
```

## pc.js

```
var server_url = "http://linda-server.herokuapp.com";
var socket = io.connect(server_url);
var linda = new Linda().connect(socket);
var ts = linda.tuplespace("korin");
linda.io.on("connect", function(){});

function move_right(){
  ts.write({type: "move", cmd: "right"});
}

function move_left(){
  ts.write({type: "move", cmd: "left"});
}

function move_stop(){
  ts.write({type: "move", cmd: "stop"});
}
```

## 付録B モータ制御回路

### 回路図

