Control of a Trailer

How to steer a car when parking

Alfredo Hernández David Masip Martí Municoy
Jan-Hendrik Niemann

Modelling for Science and Engineering

2nd February 2018

Modelling the Kinematics I

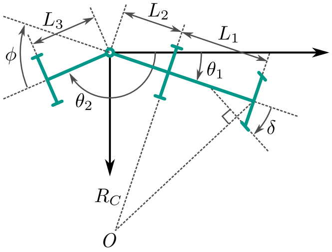


Figure: Geometric model of the car-trailer system

Parameters	Description
θ_1	Angle of the car.
θ_2	Angle of the trailer.
ϕ	Angle between car and trailer.
δ	Steering angle.
V	Speed of the car.
$V_{trailer}$	Speed of the trailer.
R_C	Distance from the centre of rotation O to rear axle of the car.
L_1	Wheelbase of the car.
L_2	Overhang from the rear axle of the car to the hitch point.
L_3	Distance from the trailer axle to the hitch.

Table: Physical parameters of the system described by our model

Modelling the Kinematics II

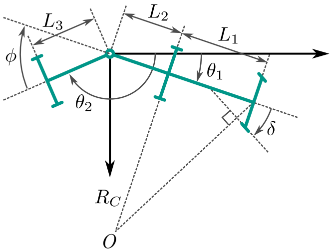


Figure: Geometric model of the car-trailer system

$$\phi = \pi - (\theta_2 - \theta_1), \quad \dot{\phi} = -\dot{\theta}_2 + \dot{\theta}_1$$

$$\dot{\theta}_1 = -\frac{V}{R_C}, \quad R_C = \frac{\tan\delta}{L_1}$$

Modelling the Kinematics II

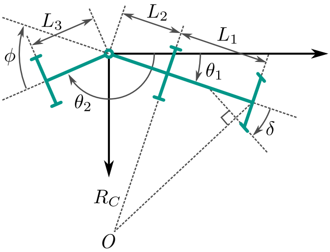


Figure: Geometric model of the car-trailer system

$$\phi = \pi - (\theta_2 - \theta_1), \quad \dot{\phi} = -\dot{\theta}_2 + \dot{\theta}_1$$

$$\dot{\theta}_1 = -\frac{V}{L_1} \tan\delta$$

Modelling the Kinematics II

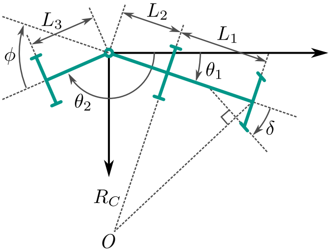


Figure: Geometric model of the car-trailer system

$$\phi = \pi - (\theta_2 - \theta_1), \quad \dot{\phi} = -\dot{\theta}_2 + \dot{\theta}_1$$

$$\dot{\theta}_1 = -\frac{V}{L_1} \tan\delta$$

$$\dot{\theta}_2 = \frac{L_2 \dot{\theta}_1 \cos\phi + V \sin\phi}{L_3}$$

Modelling the Kinematics II

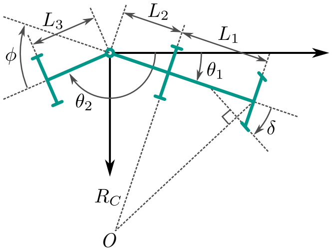


Figure: Geometric model of the car-trailer system

$$\phi = \pi - (\theta_2 - \theta_1), \quad \dot{\phi} = -\dot{\theta}_2 + \dot{\theta}_1$$

$$\dot{\theta}_1 = -\frac{V}{L_1} \tan\delta$$

$$\dot{\theta}_2 = \frac{L_2 \dot{\theta}_1 \cos\phi + V \sin\phi}{L_3}$$

$$\dot{\phi} = -\frac{V}{L_3} \sin\phi - \frac{V}{L_1} \tan\delta \left(1 + \frac{L_2}{L_3} \cos\phi\right) \tag{1}$$

Approaches to Solving the Problem

- Controller using pole placement
- Optimisation-based control
- Steering-based control

What is Control Theory?

Figure: Block diagram of a dynamical system

- Input $u(t)$ which represents controls, noises and disturbances
- Output $y(t)$ representing the measurements
- Assumption: approximation by

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)) \\ \dot{y}(t) &= h(x(t), u(t)) \end{aligned} \tag{2}$$

What is Control Theory?

- Consider system (2) is a *linear time-invariant control system* of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \dot{y}(t) &= Cx(t) + Du(t) \end{aligned} \tag{3}$$

- *System matrix* A, *input matrix* B, *output matrix* C and *feed-through matrix* D
- When a nonlinear system is given (3) is obtained by linearisation at points (x^*, u^*) such that $f(x^*, u^*) = 0$

What is Controllability?

Simply said, a system is said to be controllable if it is possible to find a control input that takes the system from any initial state to any final state in any given time interval.

Controllability: Definition - Reachable & Controllable

Definition

A system (A, B) is called

- *reachable at time $T > 0$ if for all $x^1 \in \mathbb{R}^n$ there exists $(x, u) \in \mathcal{B}_{(A,B)}$ such that $x(0) = 0$ and $x(T) = x^1$*
- *controllable at time T if for all $x^0, x^1 \in \mathbb{R}^n$ there exists $(x, u) \in \mathcal{B}_{(A,B)}$ such that $x(0) = x^0$ and $x(T) = x^1$*
- *null-controllable at time T if for all $x^0 \in \mathbb{R}^n$ there exists $(x, u) \in \mathcal{B}_{(A,B)}$ such that $x(0) = x^0$ and $x(T) = 0$.*

Controllability: Definition - Simply said

Reachable means that we can steer the system from 0 to any final state in any time Z. *Controllable* mean that it is possible to steer the system from any initial state to any final state in any given time T and hence *null-controllable* means that the final state is 0.

Controllable: Yes or No?

Theorem

For all $T > 0$ it holds

$$\mathcal{R}_0(T) = \mathrm{im}[B, AB, ..., A^{n-1}B] =: K(A, B) \in \mathbb{R}^{n \times nm}.$$

Corollary

For a system (A, B) the following statements are equivalent:

1. *There exists $T > 0$ such that (A, B) is controllable at time T.*
2. $\mathrm{im}(K(A, B)) = \mathbb{R}^n$
3. $\mathrm{rank}(K(A, B)) = n$
4. *For all $T > 0$ the system (A, B) is controllable at time T. We say that (A, B) is controllable.*

Stability

- From theory of differential equations stability of a system can be analysed by considering the eigenvalues of the matrix A
- One aim of control theory is stabilisation

Questions

What have eigenvalues and controllability in common?
How can we stabilise a given system if possible?

Stabilisation

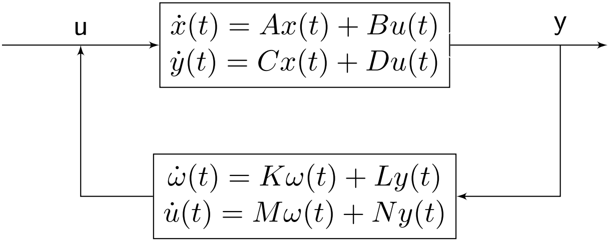


Figure: Block diagram of a feedback loop

Aim: determine a feedback controller such that

$$\begin{pmatrix} \dot{x}(t) \\ \omega(t) \end{pmatrix} = \begin{pmatrix} A + BNC & BM \\ LC & K \end{pmatrix} \begin{pmatrix} x(t) \\ \omega(t) \end{pmatrix}$$

is internally stable, i.e. all eigenvalues of A have negative real part.

Pole Placement Theorem

Theorem

Let (A, B) *be a system as in (3). The the system is controllable if and only if for all monic, i.e. the leading coefficient is 1, polynomial* $p(s) \in \mathbb{R}[s]$ *with* $\deg(p(s)) = n$ *there exists* $F \in \mathbb{R}^{m \times n}$ *such that* $p(s) = \det((A + BF) - sI)$.

Pole Placement Theorem - What does it say?

We see that if the system is controllable we can place the eigenvalues everywhere we like and therefore are able to stabilise our system.

Achieve Stability

- Linearise the model about $\phi = 0$ and $\delta = 0$ using a Taylor series:
- We get

$$\Delta\dot{\phi} = \frac{V}{L_3}\Delta\phi - \frac{V}{L_1}\left(1 + \frac{L_2}{L_3}\right)\Delta\delta \tag{4}$$

with

$$\Delta\phi = \phi - \phi_0$$

- $A = \frac{V}{L_3}$, $B = -\frac{V}{L_1}(1 + \frac{L_2}{L_3})$, $x = \Delta\phi$ and $u = \Delta\delta$
- For C we can choose 1 and therefore $y = x$ since $D = 0$

Is it controllable?

- Kalman-matrix is $K(A, B) = B$
- Since $B = -\frac{V}{L_1}(1 + \frac{L_2}{L_3}) > 0$ is a positive scalar the Kalman-matrix has full rank and therefore the system is controllable
- We can use the pole placement theorem to stabilise the system

Full-State Feedback Controller

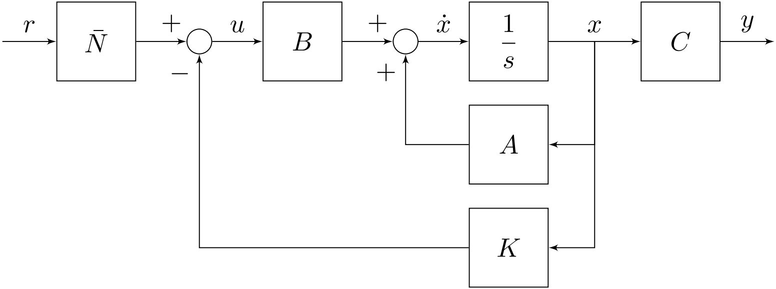


Figure: Block diagram of the state space model

Full-State Feedback Controller

For simplicity, let's assume the reference is zero, $r = 0$. The input is then

$$u = -Kx$$

The state-space equations for the closed-loop feedback system are, therefore,

$$\begin{aligned} \dot{x} &= (A - BK)x \\ y &= Cx \end{aligned} \tag{5}$$

Feedback matrix K

- Stability and time-domain performance of the closed-loop feedback system are determined primarily by the location of the eigenvalues of the matrix $(A - BK)$
- We only have 1 eigenvale to place since our system is one dimensional

$\Rightarrow$ compute such a matrix K

Eliminate Steady State Error

- $\bar{N}$ is used to scale the reference input to make it equal to Kx in steady-state $\Rightarrow$ Kx will be equal to the desired output, i.e. that we do not get a non-zero steady state error
- $\bar{N}$ is computed in general as $\bar{N} = N_u + KN_x$ where $N_x = N(1:n)$ and $N_u = N(1+n)$ using MATLAB notation and

$$N = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where $[0, ..., 0, 1]$ is a vector of size $n+1$, shortly represented as $[0, 1]$, and n the dimension of the system

Open-Loop Response: Implementation

```
% Time in seconds
t = 0:0.01:10;
% Input
u = zeros(size(t))*r;
% Create state space object
sys = ss(A,B,C,0);

% Calculate open-loop response
[y,t,x] = lsim(sys,u,t,phi_0);
```

- `ss()` defines the state space.
- `lsim()` simulates the output time response $y(t)$ of dynamic system with input $u(t)$.

Open-Loop Response: Simulation

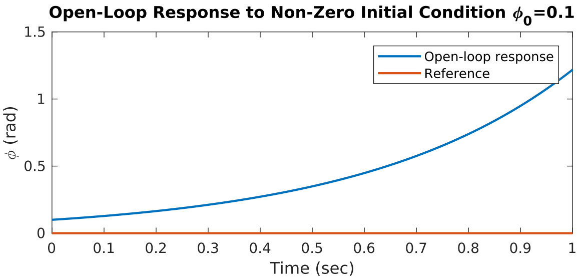


Figure: Open-Loop response to non-zero initial condition driving backwards with a trailer

Open-Loop Response: Simulation

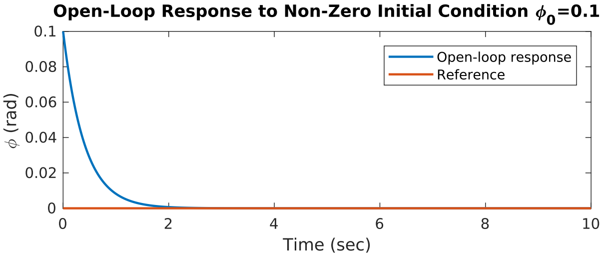


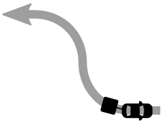
Figure: Open-Loop response to non-zero initial condition driving forwards with a trailer

Closed-Loop Response: Implementation

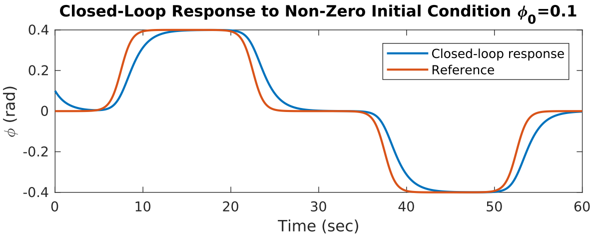
```
% Time in seconds
t = 0:0.01:60;
% Inputs
u = curve4(t, 0.4); % S shape
% Apply pole placement
K = place(A,B,-0.73);
% State space
sys_cl = ss(A-B*K,B,C,0);
% Calculate scaling factor
Nbar = rscale(sys,K);

% Calculate closed-loop response
[y_cl,t,x_cl] = lsim(sys_cl,Nbar*u,t,phi_0);
```

- `place()` finds the state-feedback gain K and provides the desired closed-loop pole.
- `rscale()` computes the desired scale factor $\bar{N}$.

Closed-Loop Response: Simulation



(a) Diagram of the curve



(b) Closed-loop response of the controller

Figure: Diagram and control of the car-trailer system in a S-shaped curve

Available at `https://github.com/aldomann/trailer-parking`

Optimisation-based

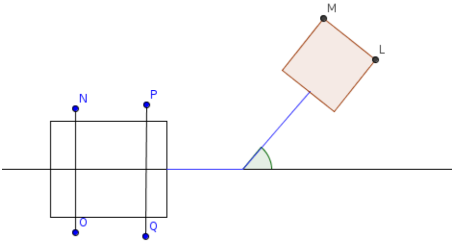


Figure: Geometric situation

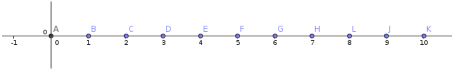


Figure: Time divisions of the method

Optimisation-based

We optimise, as a function of δ:

$$\mathrm{cost}_\alpha(y, \phi) = y^2 + \alpha\phi^2 \tag{6}$$

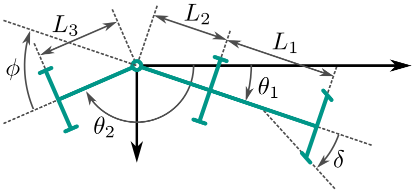
Solved using analytical expressions for the circular paths of the points in the car and using a Runge–Kutta–Fellberg 78 method to integrate the differential equation for ϕ.

Optimisation-based

Issues with the method:

- Greedy.
- Dependent on number of divisions and α.
- One variable δ for two objectives (y, ϕ).

The grounds of the approach



- Python 2.7 implementation
- Matplotlib
- NumPy

- $\dot{\phi} = \frac{V}{L_3} \cdot \sin\phi + \frac{V \cdot \tan\delta}{L_1} \cdot \left(1 + \frac{L_2 \cdot \cos\phi}{L_3}\right)$
- $\phi_i = \phi_{i-1} + \left(\frac{\sin\phi_{i-1}}{L_3} + \frac{\tan\delta}{L_1} \cdot \left(1 + \frac{L_2 \cdot \cos\phi_{i-1}}{L_3}\right)\right) \cdot V \cdot \Delta t$

2 main goals:

- Visualise and check the behaviour of the model
- Create a steering-based controller

First Tests of the Model

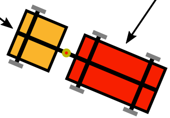
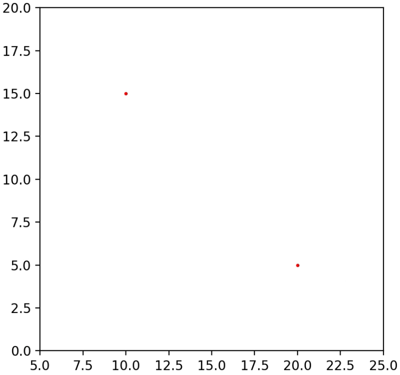
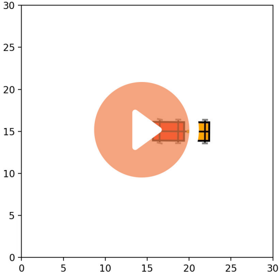


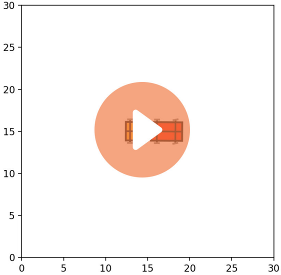
Figure: Diagram of the system

- Go from A to B backwards.



First Tests of the Model



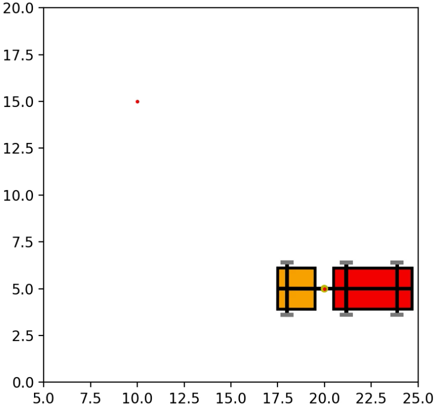


Steering-based Control

(i) Face trailer to a "non-annoying" direction.

(ii) Move the car to the goal point.

- Keep repeating (i) until reaching point B.

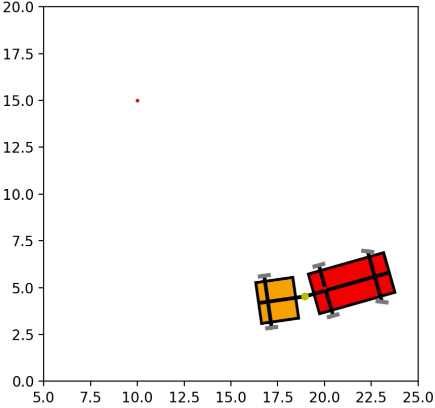


Steering-based Control

(i) Face trailer to a "non-annoying" direction.

(ii) Move the car to the goal point.

- Keep repeating (i) until reaching point B.

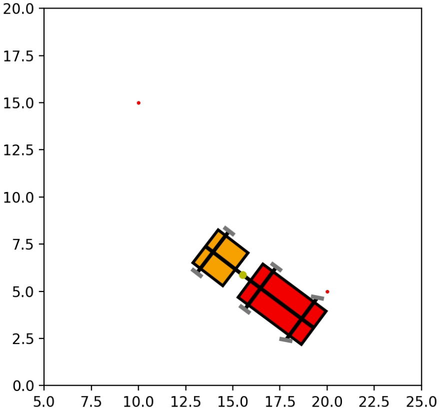


Steering-based Control

(i) Face trailer to a "non-annoying" direction.

(ii) Move the car to the goal point.

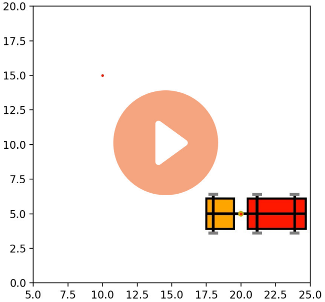
- Keep repeating (i) until reaching point B.



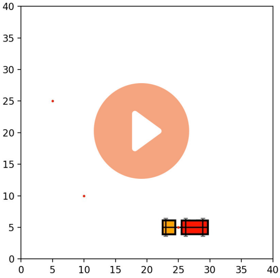
Steering-based Control

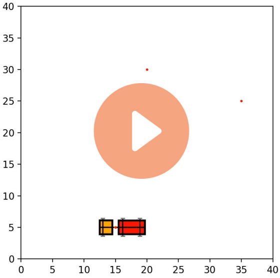
(i) Face trailer to a "non-annoying" direction.
(ii) Move the car to the goal point.

- Keep repeating (i) until reaching point B.



Steering-based Control





Available at https://github.com/martimunicoy/TrailerController

Conclusions

We saw three different approaches:

- Controller using pole placement (using MATLAB).
- Optimisation-based control (using C).
- Steering-based control (using Python).

All of them are viable options for control with their own advantages and disadvantages.

References

- J. Nilsson and S. Abraham. *Trailer Parking Assist (TPA)*. Master's Thesis, 2013.
- R. N. Jazar. *Vehicle Dynamics: Theory and Application*. Springer-Verlag New York, 2014.
- J. Chiu and A. Goswami. The critical hitch angle for jackknife avoidance during slow backing up of vehicle–trailer systems. 26, 2014.
- E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Texts in Applied Mathematics. Springer, New York, 1998.
- D. Hinrichsen and A. Pritchard. *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*. Texts in Applied Mathematics. Springer, 2005.
- H. Trentelman, A. A. Stoorvogel and M. Hautus. *Control Theory for Linear Systems*. Communications and Control Engineering. Springer London, 2001.