# Outline Thesis Feedback Control and Backpressure

* Title page
  + Title of the thesis
  + Date of the thesis defense
  + Name and student number
  + Chosen specialization
  + Master program
  + Faculty EEMCS, TU Delft
* Preface
  + Explain the topic and context (institute/company)
  + Main findings in a few lines
  + Names of the members of the thesis committee
  + Acknowledgements
  + Finish with ‘name’ and ‘date’
* Introduction
  + Main research themes
  + Overview of the report
* Previous work
  + Relevant scientific publications
  + Reformulates the research questions to comply with the state of the art in the field
* Performed work
  + Several chapters
* Discussion of the results
  + Discussion
  + Conclusion
  + Recommendations and suggestions for future research
* Appendices
  + Scientific paper written about this work

# Actual content

* Problem statement
  + Backpressure
    - What is it?
    - Why does it occur?
  + When does it become a problem
    - General problem
      * …
      * Hot/cold observables
    - Examples of backpressure being a problem
      * [Fast producer, slow consumer]
      * Zip
      * …
  + Different approaches to solve this
    - Within Rx/ReactiveStreams/etc
      * See:
        + <https://xgrommx.github.io/rx-book/content/getting_started_with_rxjs/creating_and_querying_observable_sequences/backpressure.html>
        + <https://github.com/ReactiveX/RxJava/wiki/Backpressure>
        + <http://www.reactive-streams.org/>
        + <http://www.reactivemanifesto.org/glossary#Back-Pressure>
      * Lossy (packets being dropped)
        + Debounce (throttleWithTimeout)
        + Throttling (throttleFirst)
        + Sampling (throttleLast)
        + Pausable Observable in RxJS
      * Loss-less
        + Buffers and Windows (+ complex buffer constructs)
        + Pausable buffers in RxJS
        + Controlled Observables

Reactive pull 🡪 ReactiveStreams

* + - Transmission Control Protocol
      * <https://en.wikipedia.org/wiki/Transmission_Control_Protocol#Flow_control>
* Feedback Control
  + Introduction 🡪 reference to blog, Hellerstein and Janert
    - Basics of feedback control
    - Difference mathematical approach and approach of Hellerstein/Janert
  + Introduce the ball tracker as a toy example
    - Use this throughout the rest of the Feedback Control section
  + Feedback control as ‘working with streams’
    - A component sits in between 2 streams and performs some sort of transformation
    - A component is compositional: connecting components, making feedback loop, zipping 🡪 a feedback system is the same as a component
    - Observation: ***a component is the same as a Mealy Machine***
    - Derive the exact type of a component, starting from a Mealy Machine and using category theory
    - Introduce the operators on Component
      * Concat
      * Map/peek
      * Zip
      * Feedback
      * Lift (as generalizing over all operators)
* Solving backpressure with feedback control
  + General solution
  + Solution to specific examples
  + …

Old notes on backpressure case study:

* Backpressure
  + Pull solution – control queue length
  + Push solution – control number of workers with metrics:
    - Queue length
    - Net queue length change
    - In/out ratio per timeunit