



Carsten Knoll, Robert Heedt Faculty of Electrical and Computer Engineering, Institute of Control Theory

### "Automatic Control Knowledge Repository" – A Computational Approach for Simpler and More Robust Reproducibility of Results in Control Theory

ICSTCC, Sinaia, Romania (remote participation), 2020-10-08

# Reproduction of Scientific Results (1)

- Scientific communication: traditionally based on natural language (+ mathematics + diagrams)
- Growing importance of computational methods
  - Simulations
  - Numerical linear algebra
  - Symbolic calculations with CAS

Proceedings in Applied Mathematics and Mechanics, 6/10/2020

#### Trajectory Planning for Closed Kinematic Chains Applied to Cooperative Motions in Health Care

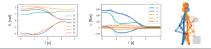
#### Carsten Knoll1\*, Xuehua Jia1, and Robert Heedt1

Physical interaction of humans can be a challenging. For uson practically relevant intuition a reasonable model is given by a local literative that on a planar right both mechanisms. From a counder perspective a counder a gama right both standing, eq., can be seen as an optimal counted problem (GPU) with input and state restrictions. Due to the algebraic constraints and the comparable planameter of pairs increased, perpeter formatication of oxida a problem in instrini. In this commonlow planetics, a strained and the comparable planameter of pairs increased, perpeter formatication of stoch a problem in notivitia. In this commonlow planetics, and the strain planetic plane

Copyright line will be provided by the publisher

#### 1 Introduction: Modelling of Cooperative Standing-Up Motion

Motivated  $a_{2}$ , by [1] and references therein we consider the basis mechanical appects of cooperairs estanting-up notion, which take place  $a_{2}$ , by  $a_{1}$  particule cars. The goal of this coordination is not be provider radius unders but radius of  $b_{11}$  and equivalent takes the state of the state state of the state state of the state state of the state the state of the





"Automatic Control Knowledge Repository" Institute of Control Theory - C. Knoll, R. Heedt ICSTCC, Sinaia, Romania (remote participation), 2020-10-08

Slide 2 of 18



# Reproduction of Scientific Results (1)

- Scientific communication: traditionally based on natural language (+ mathematics + diagrams)
- Growing importance of computational methods
  - Simulations
  - Numerical linear algebra
  - Symbolic calculations with CAS
- Understanding and reproduction depends on access to these methods (sources and executables)

Proceedings in Applied Mathematics and Mechanics, 6/10/2020

#### Trajectory Planning for Closed Kinematic Chains Applied to Cooperative Motions in Health Care

#### Carsten Knoll1\*, Xuehua Jia1, and Robert Heedt1

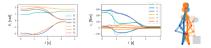
<sup>1</sup> Institut f
ür Regelungs- und Steuerungstheorie, Fakult
ät Elektrotechnik und Informationstechnik, Technische Universit
ät Dresden, 01062 Dresden

Physical interaction of humans can be a challenging. For usone practically relevant situations a reasonable model in given by a local kinemic due to a planar right both mechanism. From a courded perspective a couperative motion, e. g. assisted standing, e.g., can be seen as an equinal coursel problem (GCP) with mpart and state restrictions. Due to the algebraic constraints and the comparable planumeter of plans involved, perpet formalization of oxida a problem in activiti. In this commbulos we compare two different modeling approaches: discretization of the full dynamical problem and iterative solutions of constraints studing regions. The latter turns can be built applicationly faster while III providing — in some seme – optimal solutions.

Copyright line will be provided by the publisher

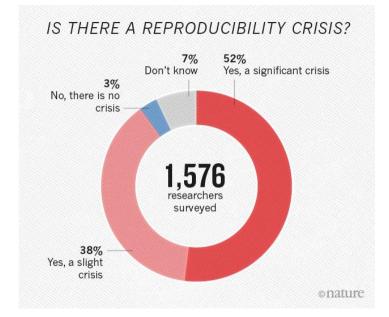
#### 1 Introduction: Modelling of Cooperative Standing-Up Motion

Motivated  $a_{2}$ , by [1] and references therein we consider the basic mechanical appects of cooperative standing-up notion, which take place  $a_{2}$ , be p (ratio) carrs. Begin of the coordivation is not provide relation unders but rather a methodergy to simplify further research. Following [1] we model each of the two persons (are; plow and receiver) a chains of  $h_{1-2}$ are considered by the standard of the coordivation is not provide relation. The standard of the two genesis is not provide the standard of the two persons (are the standard of the two persons (are the standard of the two genesis is not provide the standard of the standard of the two persons (are the standard of the standard The connection of the two kinematic chains ( $b_{1-2}$   $b_{2-1}$ ) of  $b_{2-1}$ . The standard of the standard









#### source: https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970 (2016)



source: https://www.nature.com/articles/467775a/box/2 (2010)

# Outline

#### ✓ Motivation: Reproducibility Challenge

- □ Existing Approaches
- □ Our Approach
- □ Implementation Status & Demo
- □ Conclusion & Outlook

# Outline

Motivation: Reproducibility Challenge
 Existing Approaches
 Our Approach
 Implementation Status & Demo
 Conclusion & Outlook

 DVCS: Keeping track of changes (Who? When? What?)



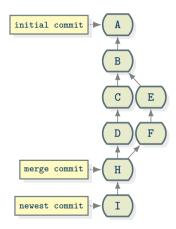


- DVCS: Keeping track of changes (Who? When? What?)
- Defacto standard: git repository





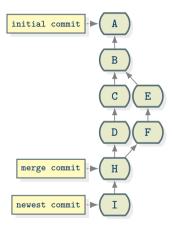
- DVCS: Keeping track of changes (Who? When? What?)
- Defacto standard: git repository
- Incremental snapshots of working directory +meta data
- stored in a hash tree







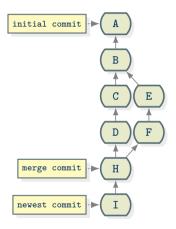
- DVCS: Keeping track of changes (Who? When? What?)
- Defacto standard: git repository
- Incremental snapshots of working directory +meta data
- stored in a hash tree
- Allows temporary parallelism (*fork* → *merge*)







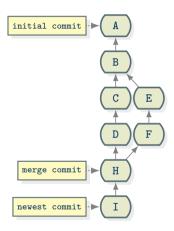
- DVCS: Keeping track of changes (Who? When? What?)
- Defacto standard: git repository
- Incremental snapshots of working directory +meta data
- stored in a hash tree
- Allows temporary parallelism (*fork* → *merge*)
- Mechanisms to resolve merge conflicts







- DVCS: Keeping track of changes (Who? When? What?)
- Defacto standard: git repository
- Incremental snapshots of working directory +meta data
- stored in a hash tree
- Allows temporary parallelism (*fork* → *merge*)
- Mechanisms to resolve merge conflicts
- ∃ platforms for public repository-hosting (github, gitlab, codeberg, ... )







- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results





- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results
- (-) Unproductive code (additional effort)





- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results
- (-) Unproductive code (additional effort)
- (+) Identify unwanted behavior (bugs)



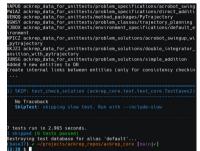


- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results
- (-) Unproductive code (additional effort)
- (+) Identify unwanted behavior (bugs)
- (+) Support different execution environments (operating systems, library versions, ...)





- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results
- (-) Unproductive code (additional effort)
- (+) Identify unwanted behavior (bugs)
- (+) Support different execution environments (operating systems, library versions, ...)
  - $\Rightarrow$   $\exists$  projects with many test cases (  $10^1 \dots 10^3$  )







- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results
- (-) Unproductive code (additional effort)
- (+) Identify unwanted behavior (bugs)
- (+) Support different execution environments (operating systems, library versions, ...)
  - $\Rightarrow$   $\exists$  projects with many test cases (  $10^1 \dots 10^3$  )

UENQQ ackrep_data_for_unittests/method_packages/PyTrajectory
QGWSY ackrep_data_for_unittests/problem_classes/trajectory_planning
YJBOX ackrep_data_for_unittests/environment_specifications/default_
ironment
<pre>HPICZ ackrep_data_for_unittests/problem_solutions/acrobot_swingup_v</pre>
UKJZI ackrep_data_for_unittests/problem_solutions/double_integrator
ansition_with_pytrajectory
JUNSG ackrep_data_for_unittests/problem_solutions/simple_addition
Added 9 new entities to DB
Create internal links between entities (only for consistency checki
<ol> <li>SKIP: test_check_solution (ackrep_core.test.test_core.TestCases)</li> </ol>
No Traceback
SkipTest: skipping slow test. Run withinclude-slow
7 tests run in 2.965 seconds.

base for alias 'default'...

unittoete/problem enecificatione/acrobot

• Continuous Integration:

Dedicated service to automated test running and result handling



Slide 7 of 18



- Automated Test: Software that runs other software
- Compare result to (hardcoded) expected results
- (-) Unproductive code (additional effort)
- (+) Identify unwanted behavior (bugs)
- (+) Support different execution environments (operating systems, library versions, ...)
  - $\Rightarrow$   $\exists$  projects with many test cases ( $10^1 \dots 10^3$ )

	<pre>ackrep_data_for_unittests/problem_specifications/acrobot_swing ackrep_data_for_unittests/problem_specifications/direct_addit</pre>
	ackrep_data_for_unittests/method_packages/PyTrajectory
QGWSY	ackrep_data_for_unittests/problem_classes/trajectory_planning
<b>YJBOX</b>	ackrep data for unittests/environment specifications/default e
ironm	ent
	ackrep_data_for_unittests/problem_solutions/acrobot_swingup_w
	aiectory
	ackrep_data_for_unittests/problem_solutions/double_integrator.
	ion_with_pytrajectory
	ackrep_data_for_unittests/problem_solutions/simple_addition
	9 new entities to DB
Creat	e internal links between entities (only for consistency checki
Ma	Traceback
	ipTest: skipping slow test. Run withinclude-slow
	tprest: skipping slow test, kun withinclude-slow
	ts run in 2.965 seconds.
	pped (6 tests passed)
	oving test database for alias 'default'
	37) / ~/projects/ackrep_repos/ackrep_core [main]/]
10.00	and a sub-offector ackiep_repair ackiep_core [matrix]

• Continuous Integration:

Dedicated service to automated test running and result handling

• Triggered e.g. by a new commit to the main branch of a repo





• Low prevalence of source code publication together with scientific results





- Low prevalence of source code publication together with scientific results
- Even lower prevalence of CI usage





- Low prevalence of source code publication together with scientific results
- Even lower prevalence of CI usage





- Low prevalence of source code publication together with scientific results
- Even lower prevalence of CI usage

Plausible reason:

- Target audience: (general) software projects
- Lack of structure and functionality specific to computational science





- Low prevalence of source code publication together with scientific results
- Even lower prevalence of CI usage

Plausible reason:

- Target audience: (general) software projects
- Lack of structure and functionality specific to computational science





- Low prevalence of source code publication together with scientific results
- Even lower prevalence of CI usage

Plausible reason:

- Target audience: (general) software projects
- Lack of structure and functionality specific to computational science

Research question:

What kind of science specific structure and functionality is suited to mitigate the reproducibility crisis?







# Outline

- ✓ Motivation: Reproducibility Challenge
- ✓ Existing Approaches
- □ Our Approach
- □ Implementation Status & Demo
- □ Conclusion & Outlook

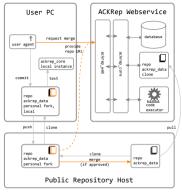
- Establish a (canonical) repository for
  - content-related software (e.g. control algorithms),
  - organizational software (automated tests)
  - and metadata in a formal structure







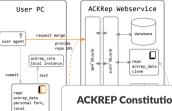
- Establish a (canonical) repository for
  - content-related software (e.g. control algorithms),
  - organizational software (automated tests)
  - and metadata in a formal structure
- Establish supportive tools and a web service for that repository
  - providing CI functionality tailored to its structure and to the needs of systems and control engineering







- Establish a (canonical) repository for
  - content-related software (e.g. control algorithms),
  - organizational software (automated tests)
  - and metadata in a formal structure
- Establish supportive tools and a web service for that repository
  - providing CI functionality tailored to its structure \_ and to the needs of systems and control engineering
- Establish a set of self-governance rules •
  - (Distribute the decision making among active contributors)



#### **ACKREP** Constitution

#### Preamble

nush clone

ackrep data

personal fork

Public

This document describes how individuals interact in the scope of the project "Automatic Control Knowledge Repository" (ACKREP). This project only makes sense if people are willing to contribute their work. Therefore a focus on equal participation and maintaining a constructive and welcoming atmosphere is central. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

#### Decision making (scalable rejection-minimizing consensus mechanism)

All decisions concerning the project are taken by the whole group of recognized contributors in form of a noll. All contributors have the right to suggest a poll, to add poll options during the discussion phase and to vote on a poll (during the voting phase). The poll allows to express approval (+1, +2), neutrality (0), or rejection (-1, -2, -3) for each poll option, Approval and rejection are recorded separately (i.e. they do not compensate). The lowest value (-2) is considered as veto. The winning option is the one with the weakest rejection, among those without yeto. In case of ambiguity stronger approval rate is relevant. If ambiguity persists, the poll is repeated with just the ambiguous options. If there is no option without veto, then

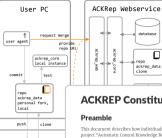




- Establish a (canonical) repository for
  - content-related software (e.g. control algorithms),
  - organizational software (automated tests)
  - and metadata in a formal structure
- Establish supportive tools and a web service for that repository
  - providing CI functionality tailored to its structure \_ and to the needs of systems and control engineering
- Establish a set of self-governance rules
  - (Distribute the decision making among active contributors)
- Working title: "Automatic Control Knowledge Repository" (ackrep)



"Automatic Control Knowledge Repository" Institute of Control Theory - C Knoll R Heedt ICSTCC, Sinaia, Romania (remote participation), 2020-10-08



ackrep data

nersonal fork

Public

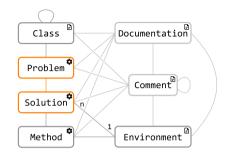
#### **ACKREP Constitution**

This document describes how individuals interact in the scope of the project "Automatic Control Knowledge Repository" (ACKREP). This project only makes sense if people are willing to contribute their work. Therefore a focus on equal participation and maintaining a constructive and welcoming atmosphere is central. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

#### Decision making (scalable rejection-minimizing consensus mechanism)

All decisions concerning the project are taken by the whole group of recognized contributors in form of a noll. All contributors have the right to suggest a poll, to add poll options during the discussion phase and to vote on a poll (during the voting phase). The poll allows to express approval (+1, +2), neutrality (0), or rejection (-1, -2, -3) for each poll option, Approval and rejection are recorded separately (i.e. they do not compensate). The lowest value (-2) is considered as veto. The winning option is the one with the weakest rejection, among those without yeto. In case of ambiguity stronger approval rate is relevant. If ambiguity persists, the poll is repeated with just the ambiguous options. If there is no option without veto, then



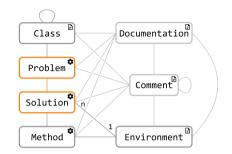






Reproducibility  $\rightarrow$  entity types:

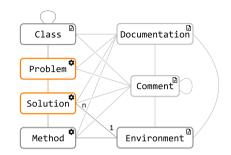
• Is a **Problem** solved by a **Solution**?







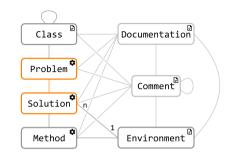
- Is a **Problem** solved by a **Solution**?
  - ProblemSpecification (code)
  - ProblemSolution (code)







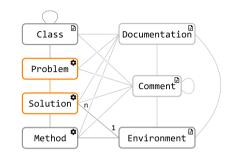
- Is a **Problem** solved by a **Solution**?
  - ProblemSpecification (code)
  - ProblemSolution (code)
- What method(s) does the solution code use?
  - MethodPackage (code)







- Is a **Problem** solved by a **Solution**?
  - ProblemSpecification (code)
  - ProblemSolution (code)
- What method(s) does the solution code use?
  - MethodPackage (code)
- Which computational **environment** is required?
  - EnvironmentSpecification (meta data)



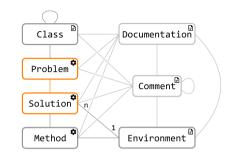




## **Proposed Repository Structure**

Reproducibility  $\rightarrow$  entity types:

- Is a **Problem** solved by a **Solution**?
  - ProblemSpecification (code)
  - ProblemSolution (code)
- What method(s) does the solution code use?
  - MethodPackage (code)
- Which computational **environment** is required?
  - EnvironmentSpecification (meta data)
- To which **class** does a Problem belong?
  - ProblemClass (meta data)



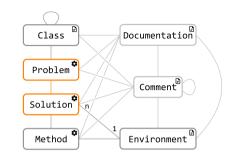




## **Proposed Repository Structure**

Reproducibility  $\rightarrow$  entity types:

- Is a **Problem** solved by a **Solution**?
  - ProblemSpecification (code)
  - ProblemSolution (code)
- What method(s) does the solution code use?
  - MethodPackage (code)
- Which computational **environment** is required?
  - EnvironmentSpecification (meta data)
- To which **class** does a Problem belong?
  - ProblemClass (meta data)
- Which further information exists?
  - Documentation (text)
  - Comment (text)



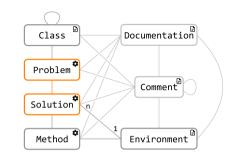




## **Proposed Repository Structure**

Reproducibility  $\rightarrow$  entity types:

- Is a **Problem** solved by a **Solution**?
  - ProblemSpecification (code)
  - ProblemSolution (code)
- What method(s) does the solution code use?
  - MethodPackage (code)
- Which computational **environment** is required?
  - EnvironmentSpecification (meta data)
- To which **class** does a Problem belong?
  - ProblemClass (meta data)
- Which further information exists?
  - Documentation (text)
  - Comment (text)



Entities:

- Identified via unique keys
- Stored in (text-based) files
- Managed via DVCS (git)
- Publicly hosted



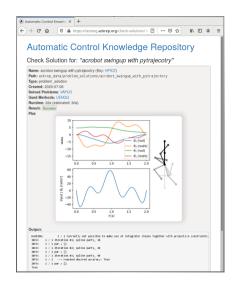




## Supportive Webservice and Tools

#### • Webservice:

- Execute Solutions against Problems (≙ Continuous Integration)
- Present results
- Make available knowledge accessible

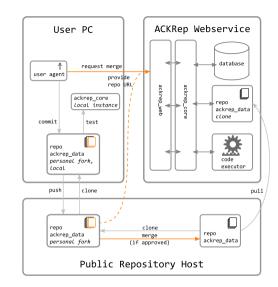






## **Supportive Webservice and Tools**

- Webservice:
  - Execute Solutions against Problems (≙ Continuous Integration)
  - Present results
  - Make available knowledge accessible
- Tools:
  - Locally check entities
  - Support debugging process







- Desirable: makes results/implementations easily findable
- Problems: single point of failure





- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish





- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish

     wuncritical due to DVCS (open sources of data and software)





- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish

     wuncritical due to DVCS (open sources of data and software)
  - Personal: bad management (rejected merge-request etc.)





Canonical repository + webservice  $\rightarrow$  central infrastructure

- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish
     w uncritical due to DVCS (open sources of data and software)
  - Personal: bad management (rejected merge-request etc.)
- Solution approach: **peer-based decision making**

#### **ACKREP Constitution**

#### Preamble

This document describes how individuals interact in the scope of the project "Automatic Control Knowledge Repository" (ACKREP). This project only makes sense if people are willing to contribute their work. Therefore a focus or equal participation and maintaining a constructive and wetcoming atmosphere is estimat. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

### Decision making (scalable rejection-minimizing consensus mechanism)

All decisions encoursing the point-rate taken by the whole group of encoginated contributions in horm of a goal all contributions have been height to compare the point of the second second second second second second on a point dividing the vector galaxies. The point allows to sequences approved a second second second second second second second second second and rejection are recorded separatively (a. the vdo nose compression). The second secon





Canonical repository + webservice  $\rightarrow$  central infrastructure

- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish
     w uncritical due to DVCS (open sources of data and software)
  - Personal: bad management (rejected merge-request etc.)
- Solution approach: peer-based decision making
  - ... about contributions (merge-requests)
  - ... about the rules themselves

#### **ACKREP Constitution**

#### Preamble

This document describes how individuals interact in the scope of the project "Automatic Control Knowledge Repository" (ACKREP). This project only makes sense if people are willing to contribute the bit work. Therefore a focus or equal participation and maintaining a constructive and welcoming atmosphere is exertical. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or as mail group.

### Decision making (scalable rejection-minimizing consensus mechanism)

All decisions encoursing the point-rate taken by the whole group of encoginated contributions in horm of a goal all contributions have been height to compare the point of the second second second second second second on a point dividing the vector galaxies. The point allows to sequences approved a second second second second second second second second second and rejection are recorded separatively (a. the vdo nose compression). The second secon





Canonical repository + webservice  $\rightarrow$  central infrastructure

- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish
     w uncritical due to DVCS (open sources of data and software)
  - Personal: bad management (rejected merge-request etc.)
- Solution approach: peer-based decision making
  - ... about contributions (merge-requests)
  - ... about the rules themselves
  - Entitled to vote: previous contributors

#### **ACKREP Constitution**

#### Preamble

This document describes how individuals interact in the scope of the project "Automatic Control Knowledge Repository" (ACKREP). This project only makes sense if people are willing to contribute their work. Therefore a focus or equal participation and maintaining a constructive and wetcoming atmosphere is estimat. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

### Decision making (scalable rejection-minimizing consensus mechanism)

All decisions concerning the project are taken by the whole group of comparison decisions in form of a plat. In a transitions have the height to compared contribution is been been by the plat of the plat of the one application of the plat of the plate of the plat of the plat of the plat of the plat of the plate of th





Canonical repository + webservice  $\rightarrow$  central infrastructure

- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish
     w uncritical due to DVCS (open sources of data and software)
  - Personal: bad management (rejected merge-request etc.)
- Solution approach: peer-based decision making
  - ... about contributions (merge-requests)
  - ... about the rules themselves
  - Entitled to vote: previous contributors
  - Method: "Scalable Rejection Minimizing Consensus Mechanism"

#### **ACKREP Constitution**

#### Preamble

This document describes how individuals interact in the scope of the project "Nationanic Control Stowledge Repository" (ACREP). This project only makes sense if people are willing to contribute their work. Therefore a focus or equal participation and maintaining a constructive and welcoming atmosphere is escital. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

### Decision making (scalable rejection-minimizing consensus mechanism)

All decision concerning the project are taken by the whole group of exceptioned combinations in form of a plat. But contributions have the hight to suggest a plat, in and plat options ming the influenciatory have and to work - plate the plat option of the plat option of the plat option and rejection are recorded repractively (a. they do not composite). The summary option are recorded repractively (a. they do not composite). The stronger approval rate is relevant. If antibipity persists, the roal is reproved that the relevant option of the road option without vector, then





Canonical repository + webservice  $\rightarrow$  central infrastructure

- Desirable: makes results/implementations easily findable
- Problems: single point of failure
  - Technical: public repo or webservice could vanish
     w uncritical due to DVCS (open sources of data and software)
  - Personal: bad management (rejected merge-request etc.)
- Solution approach: peer-based decision making
  - ... about contributions (merge-requests)
  - ... about the rules themselves
  - Entitled to vote: previous contributors
  - Method: "Scalable Rejection Minimizing Consensus Mechanism"

#### **ACKREP Constitution**

#### Preamble

This document describes how individuals interact in the scope of the project "Nationanic Control Stowledge Repository" (ACREP). This project only makes sense if people are willing to contribute their work. Therefore a focus or equal participation and maintaining a constructive and welcoming atmosphere is escital. The main purpose of this constitution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

### Decision making (scalable rejection-minimizing consensus mechanism)

• Fallback option: hard fork of software and data (possible due to free license)



Slide 13 of 18



# Outline

- ✓ Motivation: Reproducibility Challenge
- ✓ Existing Approaches
- ✓ Our Approach
- □ Implementation Status & Demo
- □ Conclusion & Outlook

## **Current Implementation Status**

- $\exists ackrep_core (GPLv3)$
- ∃ ackrep\_web (GPLv3)
- ∃ ackrep\_data
  - Default license GPLv3
  - Demonstration examples only
  - Control theoretic content in preparation





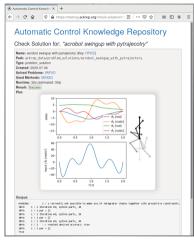
## **Current Implementation Status**

- $\exists ackrep_core (GPLv3)$
- ∃ ackrep\_web (GPLv3)
- ∃ ackrep\_data
  - Default license GPLv3
  - Demonstration examples only
  - Control theoretic content in preparation
- $\Rightarrow$  Early prototype, proof of concept





### **Live Demonstration**



#### https://testing.ackrep.org



"Automatic Control Knowledge Repository" Institute of Control Theory - C. Knoll, R. Heedt ICSTCC, Sinaia, Romania (remote participation), 2020-10-08

Slide 16 of 18

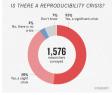


# Outline

- ✓ Motivation: Reproducibility Challenge
- ✓ Existing Approaches
- ✓ Our Approach
- ✓ Implementation Status & Demo
- □ Conclusion & Outlook

Summary:

• Reproducibility of computational results is still an unsolved challenge

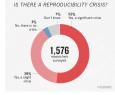






### Summary:

- Reproducibility of computational results is still an unsolved challenge
- Existing approaches (public code, CI) not widely enough used







### Summary:

- Reproducibility of computational results is still an unsolved challenge
- Existing approaches (public code, CI) not widely enough used
- Proposal: Combination of existing components
  - Structured repository





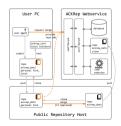




### Summary:

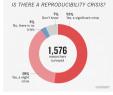
- Reproducibility of computational results is still an unsolved challenge
- Existing approaches (public code, CI) not widely enough used
- Proposal: Combination of existing components
  - Structured repository
  - Tailored webservice





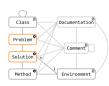


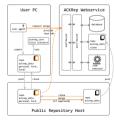




### Summary:

- Reproducibility of computational results is still an unsolved challenge
- Existing approaches (public code, CI) not widely enough used
- Proposal: Combination of existing components
  - Structured repository
  - Tailored webservice
  - Peer-based organization structure







#### **ACKREP Constitution**

#### Preamble

This document describes how individuals interact in the scope of the project" Automatic Controk Knowledge Repositosy' (ACMER). This project only makes sense if people are willing to contribute their work. Therefore a focus on equal participation and maintaining a constructive and wereforming atmosphere is excital. The main propose of this coastlution is to ensure that the project is driven by equal peers and not harmfully dominated by an individual or a small group.

#### Decision making (scalable rejection-minimizing consensus mechanism)

All decisions concerning the poier care taken by the whole group of recognized contributions in from e1 q all. In decisions have been being to suggest a quality basis to find a spin of the site of the site of the single of the site of



"Automatic Control Knowledge Repository" Institute of Control Theory - C. Knoll, R. Heedt ICSTCC, Sinaia, Romania (remote participation), 2020-10-08



### Summary:

- Reproducibility of computational results is still an unsolved challenge
- Existing approaches (public code, CI) not widely enough used
- Proposal: Combination of existing components
  - Structured repository
  - Tailored webservice
  - Peer-based organization structure

### **Outlook:**

Improve ackrep\_core (containerization), add more content to ackrep\_data







### Summary:

- Reproducibility of computational results is still an unsolved challenge
- Existing approaches (public code, CI) not widely enough used
- Proposal: Combination of existing components
  - Structured repository
  - Tailored webservice
  - Peer-based organization structure

### **Outlook:**

- Improve ackrep\_core (containerization), add more content to ackrep\_data
- Incorporate feedback from potential users/contributors
- $\rightarrow$  https://testing.ackrep.org repository: https://github.com/cknoll/ackrep\_data
- $\rightarrow$  Corresponding author: carsten.knoll@tu-dresden.de





