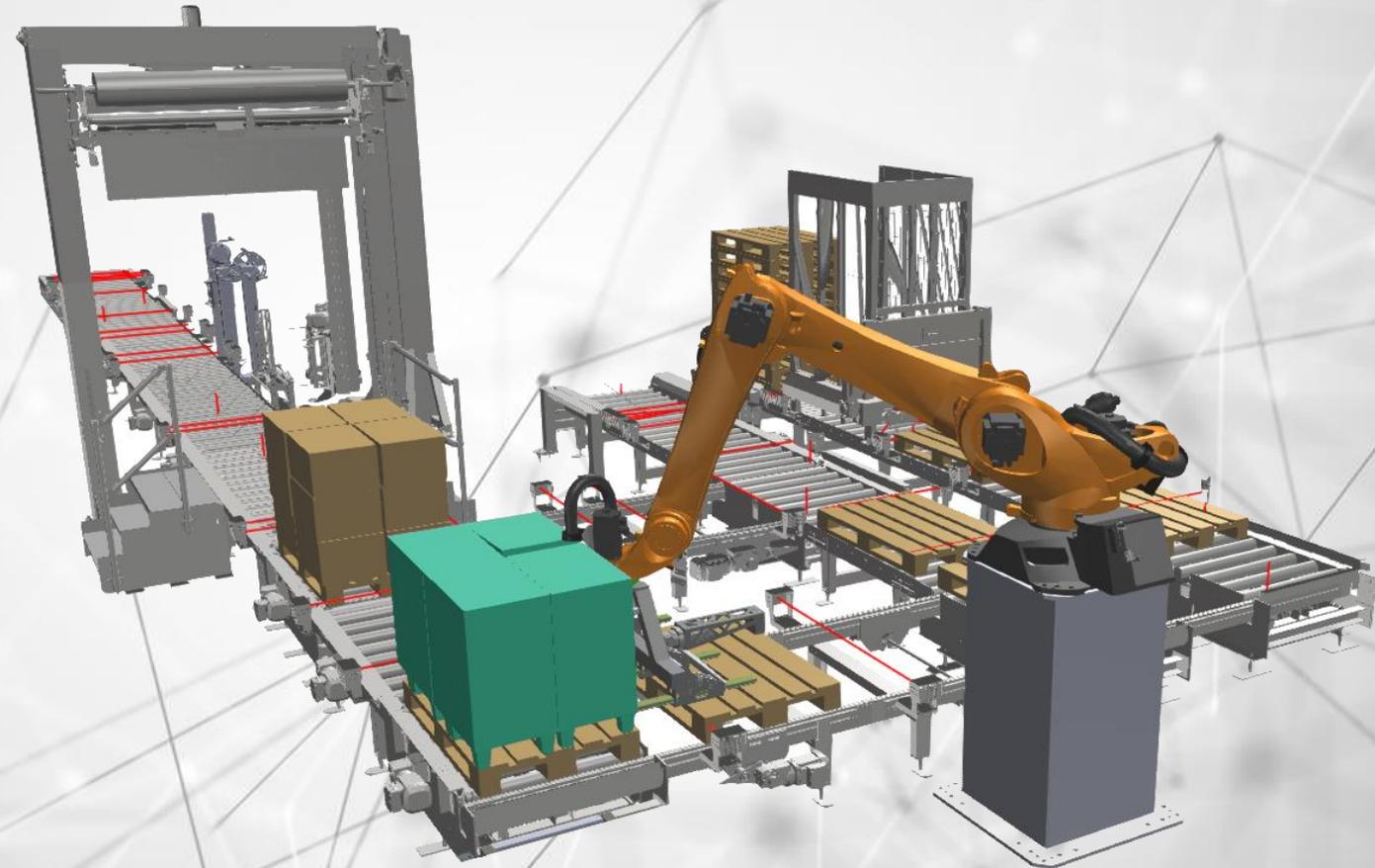


Steuerungssoftware vollständig testen – Wie digitale Zwillinge das Engineering von Maschinen revolutionieren

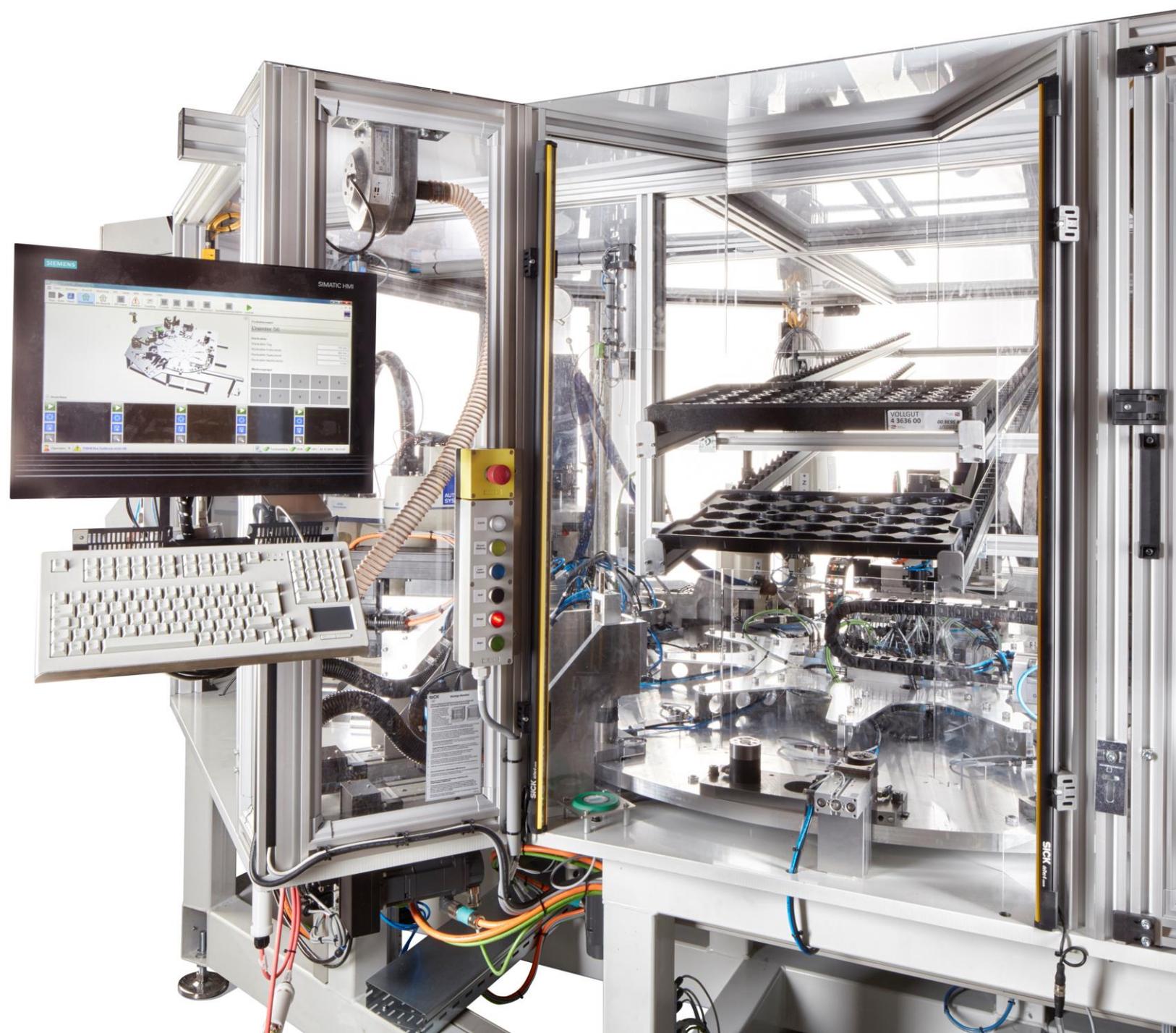
Michael Eberle
digifai

Ein Nachmittag mit Digitalen Zwillingen (TeKom)
12.05.2023, Konstanz



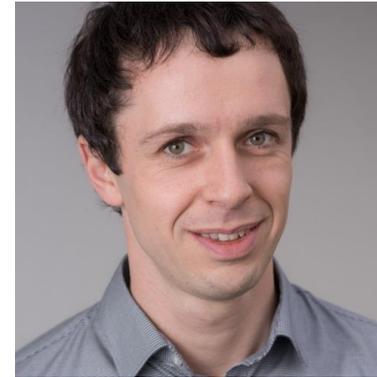
Warum braucht es eine Revolution?

Es werden doch
innovative
Maschinen
gebaut, die Ihren
Job hervorragend
erledigen!



Agenda

- Warum braucht es eine Revolution?
- Welche Rolle spielt die Physik?
- Use-Cases aus der Praxis
- Diskussion



DI Michael Eberle
Head of Modelling & Simulation

Trotzdem laufen nicht alle Projekte rosig ab

 Die Inbetriebnahme von Maschinen dauert zu lange und Personal ist schwer zu bekommen

Realistisches Testen der Steuerungssoftware

Attraktivere Arbeitsumgebung schaffen

 Konzeptionelle Fehler, auf die man doch eigentlich früher hätte drauf kommen können

Realistisches Testen von Mechanik und Abläufen

 Automatisierungsprojekte kosten zu viel

Risiko reduzieren

 Never Change a Running System

Weil ein Sicherheitsnetz fehlt!

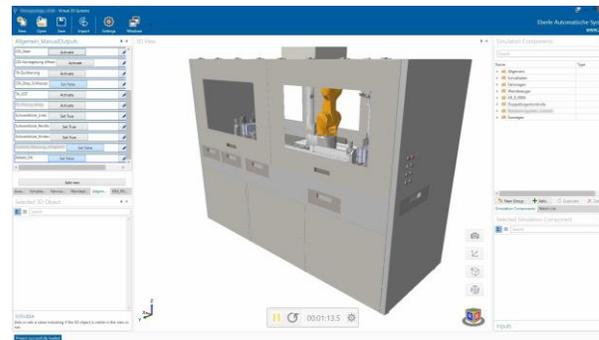
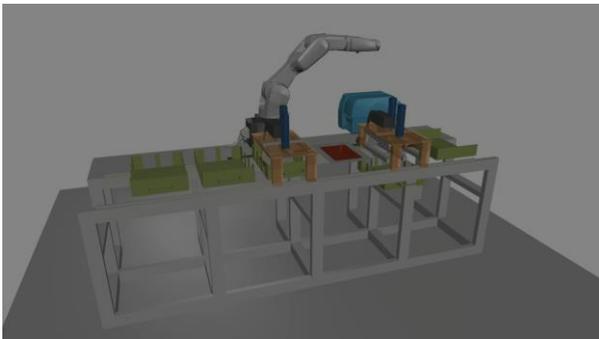
Wir von digifai haben dafür eine Lösung gesucht
und im physikbasierten digitalen Zwilling gefunden

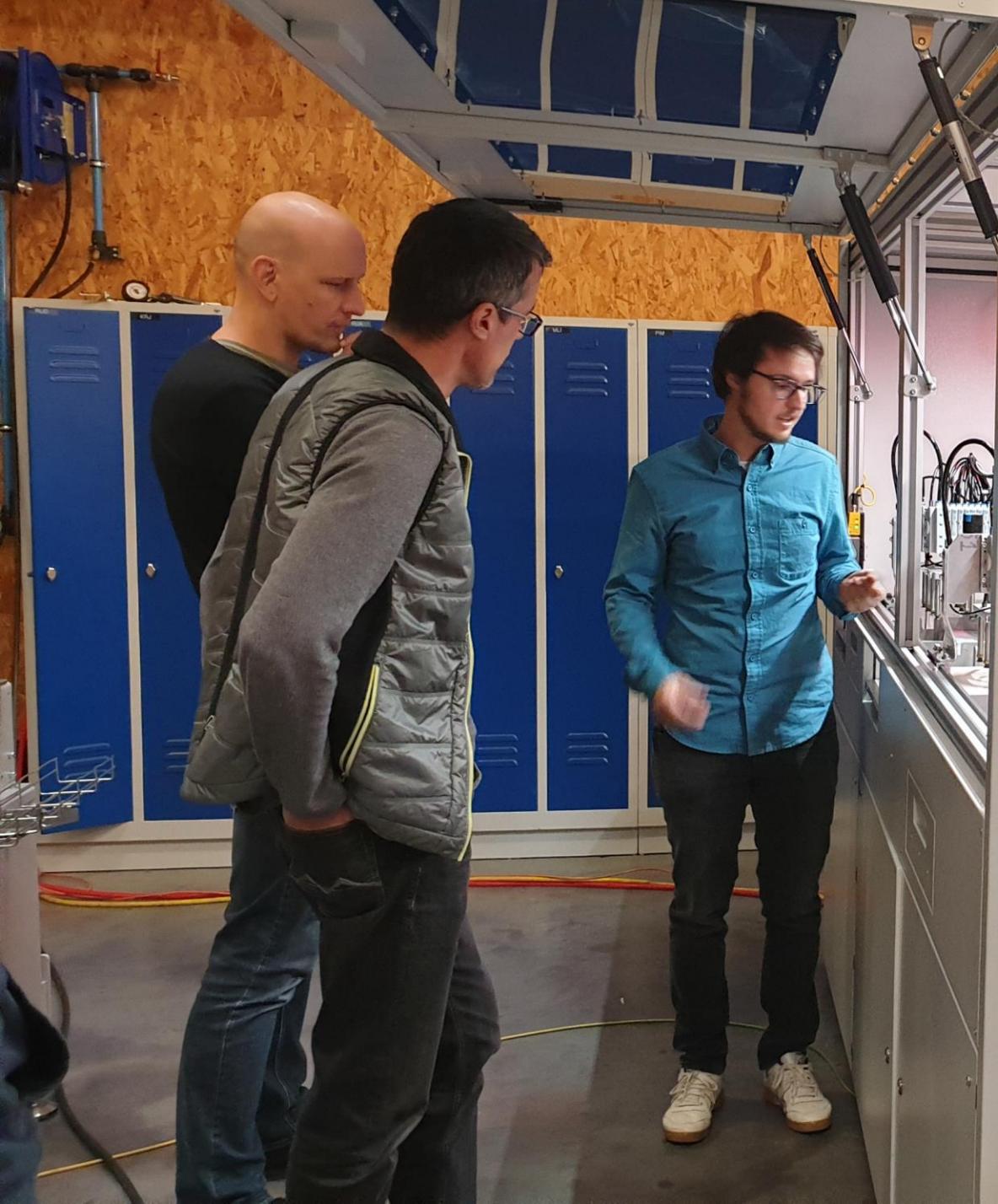
Digitale Zwillinge revolutionieren das Engineering von Maschinen

Machbarkeitsstudien
während der Konzeption
bzw. Angebotsphase

Entwickeln und Testen
von Steuerungssoftware
am digitalen Zwilling
(virtuelle
Inbetriebnahme)

Interaktives 3D-HMI an
der laufenden Maschine



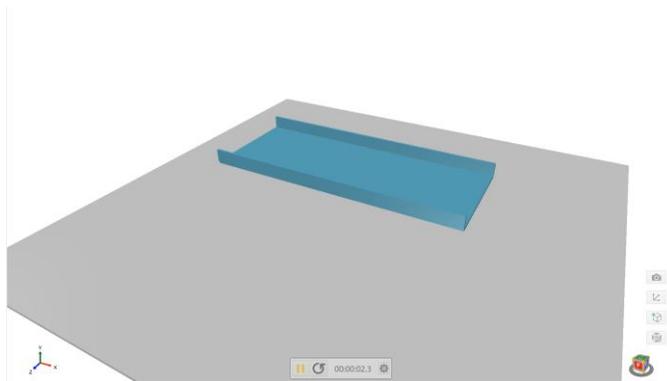


Wir entwickeln die Simulationssoftware

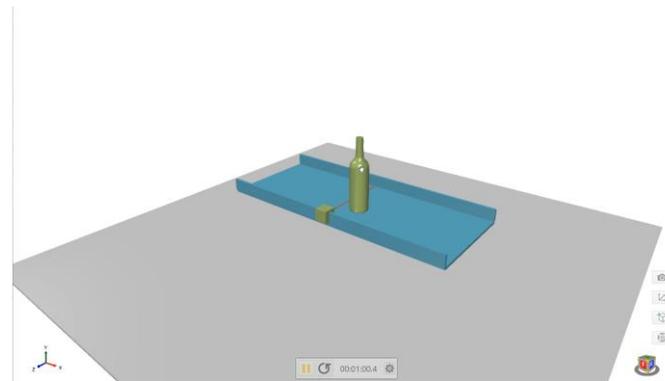
- Die Physik ist die Basis
 - 3D Simulation
 - Elektrische Vorgänge
 - Hydraulisch Vorgänge
 - Pneumatisch Vorgänge
- Unabhängig
 - vom CAD
 - vom Steuerungshersteller
- Intuitive Bedienung und Bibliotheken erleichtern die Arbeit



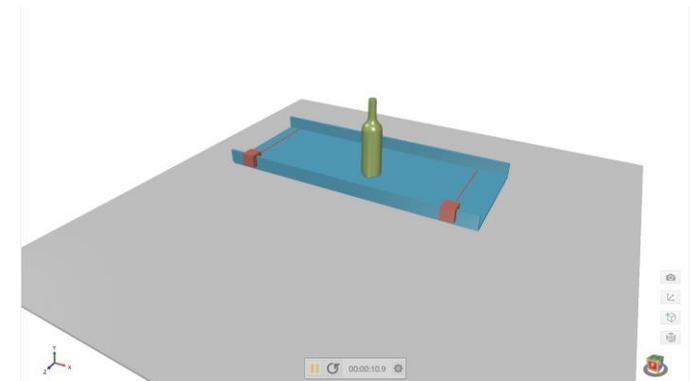
Der digitale Zwilling ist mehr als nur eine schöne 3D-Animation



Festkörpersimulation



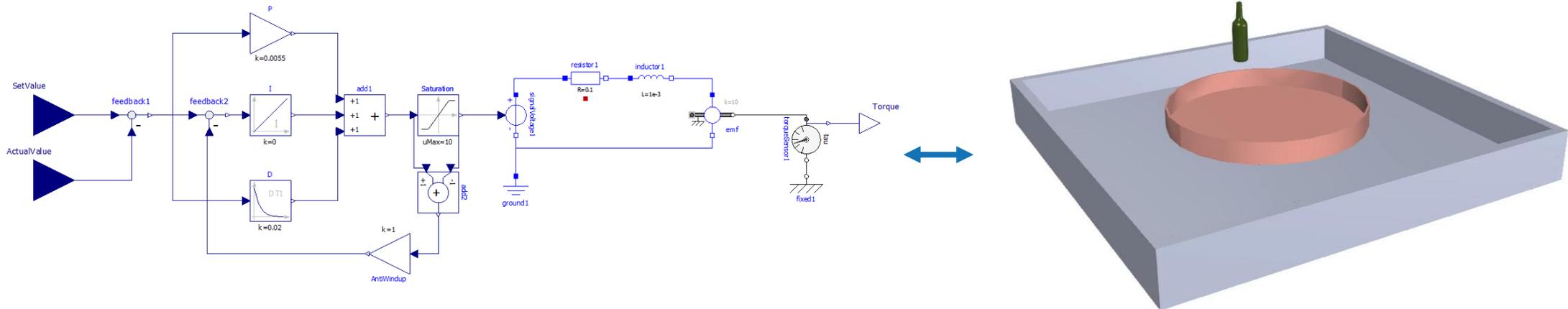
Simulation von Sensoren



Simulation von Aktoren
und Hardware bzw.
Software in the Loop

Multiphysiksimulation im digitalen Zwilling

- Die reine Festkörpersimulation genügt nicht immer
- Manche Prozesse im Maschinenbau erfordern zusätzlich eine erweiterte physikalische Modellierung
- Beispiel: Ein PID-Regler mit dem Modell eines Gleichstrommotors (counter EMF wurde vernachlässigt) regelt die Lage eines Drehtellers



Use-Cases aus der Praxis

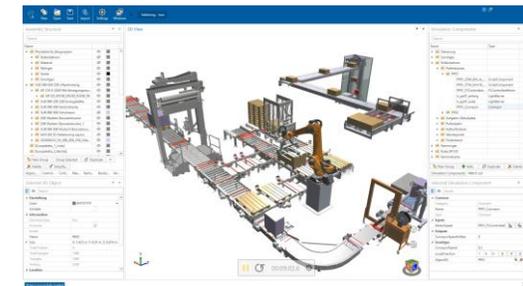
Use-Case: Der digitale Zwilling als Unterstützung im Vertrieb

- Die Präsentation komplexer Maschinen beim Kunden mit Handskizzen ist aufwändig und kann zu Missverständnissen führen
- Mit überschaubarem Aufwand können Ideen in lebende Modelle verwandelt werden
- Risiken können bereits zu Beginn besser abgeschätzt werden
 - Komponentenauswahl
 - Taktzeitanalysen
 - Bottle-Necks



Use-Case: Virtuelle Werksabnahme (FAT)

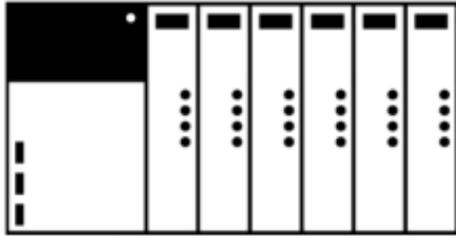
- Die Steuerungssoftware eines großen Fördertechnikprojektes wurde vollständig am digitalen Zwilling programmiert und getestet
- Während der Programmierphase konnte dem Kunden laufend Fortschritte gezeigt werden. Seine Wünsche wurden sofort umgesetzt.
- Klingen da agile Methoden in der Steuerungstechnik durch?
- Am Ende gab es einen virtuellen FAT, bei dem die Maschine vom Kunden abgenommen wurde



Use-Case: Schulung und Weiterbildung von Bedienpersonal

- Neues Bedienpersonal steht vor komplexen Maschinen
- Stillstand ist teuer
- Einfach mal was ausprobieren ist nicht möglich

Use-Case: Schulung und Weiterbildung von Bedienpersonal



SPS

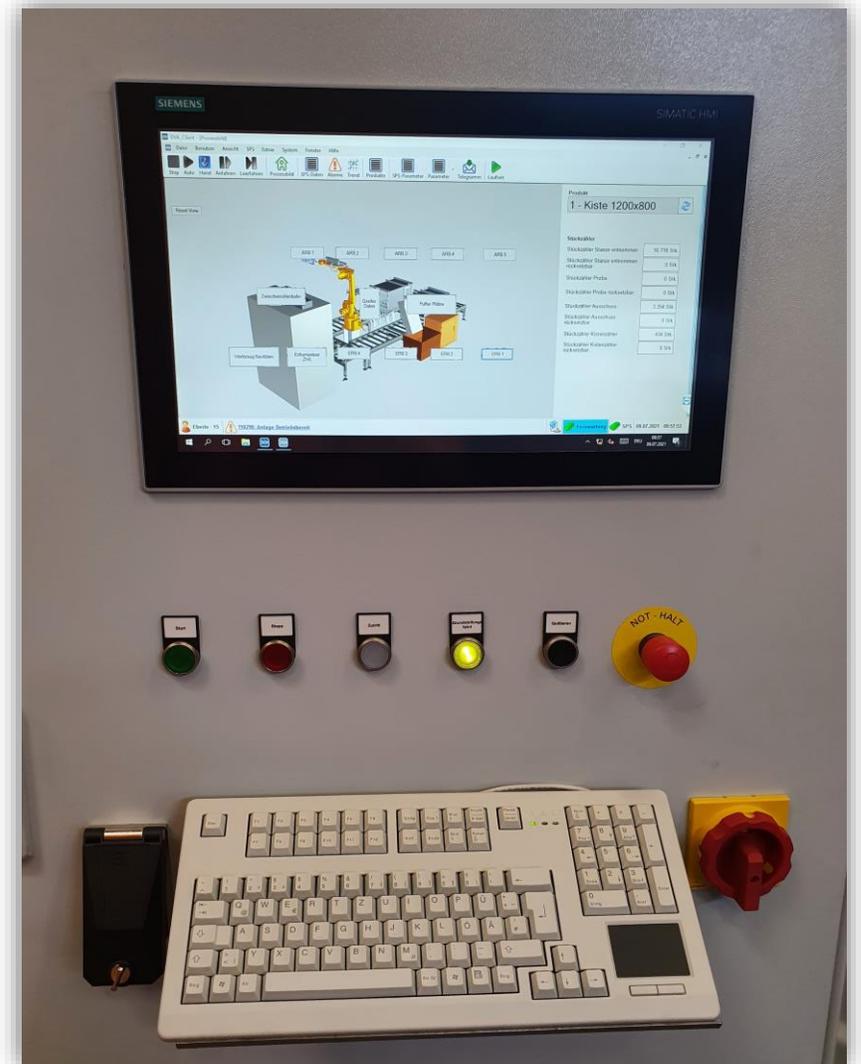


storage
(Käselagerverwaltung)



Use-Case: Intelligentes 3D-HMI an der Maschine

- Anlagenbediener fordern bessere User-Experience
- Handbetriebe sind am 3D-Modell intuitiv und einfach zu bedienen
- Alarme werden dort angezeigt, wo sie auftreten
- Support durch Techniker wird vereinfacht, da sie den Zustand der Maschine in 3D sehen



Takeaways

Die Steuerungssoftware wird in Zukunft noch mehr Aufgaben übernehmen

Zur Sicherstellung der Softwarequalität brauchen wir eine realistische Testumgebung

Maschinensimulation in 3D bildet dafür die notwendige Basis

Schön, dass Sie hier waren!

Mehr auf
www.digifai.com

Folge uns
auf LinkedIn

