

Projektskizze

Es soll ein System beschrieben, aufgebaut und untersucht werden, in dem zwei Software-Agenten mit gegenläufigen Interessen auf ein simuliertes Gasnetz einwirken.

Das gesamte System besteht aus drei Teilen. Der erste Teil ist ein einfacher Simulator für ein Gasnetz, der bereits fertiggestellt ist und genutzt werden kann. Der Simulator berechnet die zeitliche Entwicklung des inneren Zustands des Gasnetzes. Die beiden anderen Teile des Systems sind zwei Agenten.

Der erste Agent ist der Gasnetz-Dispatcher. Er kontrolliert die aktiven Elemente des Netzes (Schieber, Regler, Verdichter) und muss dafür sorgen, dass alle Drücke immer innerhalb ihrer vorgegebenen Grenzen liegen und dass die nominierten Flüsse an den Einspeiseknoten möglichst gut eingehalten werden. Druckverletzungen und Abweichungen von den nominierten Flüssen kosten Strafpunkte.

Zusätzlich kann der erste Agent Gas an von ihm gewählten Knoten kaufen oder verkaufen. In Realität entspricht dies der sogenannten Regelernergie. Kauf oder Verkauf von Regelernergie kosten Strafpunkte, deshalb sollte der Agent diese Aktionen vermeiden.

Der zweite Agent ist der Gas-Händler. Er kann im Rahmen seiner vorgegebenen Kapazitätsverträge die Nominierungen an den Randknoten des Gasnetzes vorgeben. Dabei muss er darauf achten, dass sich Einspeisungen in das Gasnetz und Ausspeisungen aus dem Gasnetz über den Tag die Waage halten.

Jeder der beiden Agenten möchte möglichst wenig Strafpunkte bekommen. Der Dispatcher-Agent bekommt Strafpunkte, wenn

- Drücke zu einem Zeitpunkt außerhalb der vorgegebenen Schranken liegen,
- Flüsse an Randknoten von den gegebenen Nominierungen abweichen,
- er Regelernergie kauft oder verkauft.

Der Händler-Agent bekommt Strafpunkte, wenn

- die Summe aller Nominierungen über einen Tag ungleich der Summe aller nicht nominierten Abnahmen im Netz ist.

Um Strafpunkte zu vermeiden, hat der Händler also dafür zu sorgen, dass über jeden Tag gemittelt genauso viel Gas ins Gasnetz fließt, wie Gas das Gasnetz verlässt.

Inhalt der Masterarbeit sollen Design und Implementierung der beiden Agenten sein. Der Gasnetzsimulator wird zur Verfügung gestellt ([github-Repo](#)) da das zusätzliche Erstellen eines Simulators den Umfang einer Masterarbeit übersteigen würde.

Ziel der Arbeit ist zu prüfen, ob es mit Hilfe von Machine Learning gelingt, einen Agenten zu bauen, der (einfache und kleine) beispielhafte Gasnetz so steuern kann, dass er sich gegenüber einem Händler behaupten kann.

Für auf Machine Learning basierende Agenten wird normalerweise eine große Menge an Trainingsdaten benötigt. Der hier gewählte Ansatz der asymmetrischen Modellierung von gleich zwei Agenten macht es unnötig, extern generierte Trainingsdaten zur Verfügung zu stellen, da die beiden Agenten gegeneinander „spielen“ und sich somit gegenseitig trainieren. Ein vergleichbarer Ansatz wurde bei Alpha Go Zero gewählt (dort allerdings symmetrisch: beide Spieler haben dieselbe Code-Basis).

Zwischenbemerkung

Es soll nicht versucht werden alle im letzten Abschnitt vorgestellten Features sofort umzusetzen. Vielmehr sind diese als Ideensammlung und mögliche Ausbaustufen zu verstehen, wenn die Arbeit sehr schnell Fortschritte macht und ausreichend Zeit zur Behandlung weiterer Aspekte zur Verfügung steht. Zu Beginn soll auf den Aufbau des Händler-Agenten verzichtet werden. Die erforderlichen Nominierungen werden automatisch zufällig generiert oder fest vorgegeben (das ist für das vorliegende Beispielnetz bereits programmiert). Der erste Agent soll sich auf die Steuerung der aktiven Elemente beschränken, der Kauf- und Verkauf von Regelenergie soll also zunächst kein Thema sein. Weitere Ausbaustufen im Rahmen der Arbeit sind möglich, sollen aber nur nach Rücksprache erfolgen und wenn noch genügend Zeit zur Verfügung steht.

Weiteres fachliches Hintergrundwissen

Der erste Agent soll einen menschlichen Dispatcher möglichst gut ersetzen. Deshalb muss die Aufgabenstellung, die er bekommt, eine möglichst gute Übersetzung der Aufgabenstellung an den menschlichen Dispatcher sein.

Was muss ein Dispatcher im Gastransport also tun und wie kann man seine Aufgabe in die Agenten-Welt übersetzen?

Ein Dispatcher muss angemeldete Gasflüsse an Einspeiseknoten stundengenau möglichst gut verwirklichen; unterstündige Abweichungen sind erlaubt (ggf. bietet sich, z.B. aus numerischen Gründen auch ein anderes Zeitintervall an). Außerdem muss er an Ausspeiseknoten vorgegebene Druckintervalle einhalten. Da Gastransportnetze in einer Verbindung miteinander stehen, führt dies zu einer Verzahnung der Pflichten der jeweiligen Dispatcher: Der vorgelagerte Dispatcher stellt ein Druckintervall zur Verfügung, mit dessen Hilfe der nachgelagerte Dispatcher den von den Kunden angemeldeten Fluss realisieren muss.

Übersetzt für unseren Agenten bedeutet dies: An den Einspeiseknoten wird ein Fluss vorgegeben, der möglichst gut einzuhalten ist. Der Agent kann sich aber darauf verlassen, dass ihm am Einspeiseknoten jederzeit ein Druck gegeben wird, mit dem die Transportaufgabe erfüllt werden kann. Der Druck kann natürlich durchaus schwanken und wird dies auch abhängig vom aktuellen Fluss tun.

An Ausspeiseknoten steht der Fluss fest. Denn die Steuerung des Flusses ist ja in Realität Aufgabe des nachgelagerten Dispatchers (der in unserem Modell perfekt arbeitet). Der Agent muss in Vertretung unseres eigenen Dispatchers lediglich dafür sorgen, dass der Druck immer im erlaubten Intervall bleibt. Abweichungen werden bestraft.

Der Simulator

Die Welt, in der die beiden Agenten arbeiten, wird durch den Simulator bereitgestellt. Der Simulator modelliert grob gesprochen vier Arten von Elementen, aus denen sich das Gasnetz zusammensetzt:

- Rohre und ihre Verbindungsknoten
- Besondere Rohrnetzelemente wie Schieber, Rückschlagklappen, Regler und Verdichter (Anm.: Regler werden über Widerstände mit variablem Widerstandsbeiwert modelliert)
- Ausspeiseknoten
- Einspeiseknoten

In unserem Fall ist der Simulator nicht direkt in einer dafür üblichen Programmiersprache programmiert, sondern er basiert auf der Optimierungssoftware Gurobi. Dem Optimierer müssen also in einer seiner Modellierungssprachen die einzelnen Elemente beigebracht werden. Die Art der Modellierung führt dazu, dass die Ergebnisse des Optimierers aussehen wie die eines Simulators. Insbesondere sind die Parameter für die aktiven Elemente keine Entscheidungsvariablen des Optimierers, sondern werden von außen vorgegeben.

Für die Masterarbeit ist ein Kapitel für die Beschreibung des Simulators wünschenswert, da eine textuelle Beschreibung bislang nicht existiert. Dazu sind die Quelltexte im Github-Repository und die Dokumente im [docs-Verzeichnis](#) des Repos zu studieren. Für Fragen und den erforderlichen Austausch stehen (fast) jederzeit Kollegen zur Verfügung.

Rohre und ihre Verbindungsknoten

Hier reicht es, die üblichen Gesetze der Gasphysik, vor allem Gleichungen für Druckabfall und Massefluss im Knoten abzubilden. Details stehen im [Paper](#) von Felix Hennings und in Skripten wie [model.py](#). Da die Entscheidungsvariablen in `model.py` von außen kommen, ist eine genauere Modellierung der Physik möglich als im Paper. Insbesondere wird anders als im Paper nicht der gesamte Betrachtungszeitraum auf einmal simuliert, sondern von Zeitschritt zu Zeitschritt gerechnet. Dabei fließen Ergebnisse des vorangehenden Zeitschritts in die Berechnung eines nachfolgenden Zeitschritts ein, etwa um Produkte zu modellieren.

Ausspeiseknoten

An Ausspeiseknoten soll der vorgegebene Fluss das Netz verlassen. Dies muss unter Einhaltung der modellierten physikalischen Gesetzmäßigkeiten geschehen. Sollte dies nicht möglich sein, wird der Optimierer entsprechende Slackvariablen nutzen, die per Zielfunktion bestraft werden, um das Modell zulässig zu machen.

An Ausspeiseknoten soll für den Steuerungsagenten bekanntlich derjenige Fluss feststehen, der das Gasnetz verlässt. Diesen Fluss zu realisieren, ist also Aufgabe des Simulators. Diese Aufgabe wird dadurch erledigt, dass der Optimierer

geeignete Flussslacks bekommt, die in der Zielfunktion bestraft werden. Er ist also per Zielfunktion daran interessiert, den Fluss möglichst gut darzustellen.

Der Optimierer bekommt an Ausspeiseknoten keine Drucksacks. Denn die Aufgabe, einen geeigneten Druck am Ausspeiseknoten zu gewährleisten, liegt ja beim Agenten und nicht beim Simulator.

Einspeiseknoten

Einspeise- und Ausspeiseknoten verhalten sich für den Agenten gewissermaßen komplementär: Wo beim einen der Fluss eingehalten werden muss, ist es beim anderen der Druck. Daraus könnte man ableiten, dass für den Optimierer analoges zu gelten hat: Bestrafung von Flussslacks am Ausspeiseknoten und daher Bestrafung von Drucksacks am Einspeiseknoten.

Ganz so einfach ist die Sache aber nicht. In Realität hängt der am Einspeiseknoten herrschende Druck oft vom Fluss ab: hoher Fluss führt zu einem niedrigen Druck, niedriger Fluss zu einem hohen Druck. Dieses Verhalten soll auch im Simulator abgebildet werden. Man könnte dies tun, indem man Drucksacks definiert, die in geeigneter Weise vom Fluss abhängen. Es gibt aber eine elegantere Möglichkeit, dies zu tun. Da über Rohre ein Druckabfall entsteht, der vom Fluss abhängig ist, haben wir vor jedem Einspeiseknoten das Gasnetz des Simulators in geeigneter Weise erweitert. Dieses zusätzliche Hilfsnetz ist nur dem Simulator bekannt, die beiden Agenten wissen davon nichts. Für sie endet das Netz am Einspeiseknoten. Für den Simulator hingegen ist dieser Einspeiseknoten lediglich ein innerer Knoten.

Vor jeden Einspeiseknoten x wird ein gleichartiges Hilfsnetz gesetzt, das im Folgenden beschrieben ist. Das Hilfsnetz hat zwei Einspeiseknoten EN_x und EH_x mit unterschiedlichen Drucksacks. An EN_x liegt der gewünschte Entry-Druck an, an EH_x 15% mehr. Über EN_x fließt maximal der vertraglich zugesicherte Fluss, über EH_x fließen maximal 20% davon.

Es gibt drei Arten von Rohren: M, G und K.

Von der Art M (Messrohr) gibt es nur ein Rohr. Es verbindet das den Agenten zugängliche Netz mit dem Hilfsnetz des jeweiligen Einspeiseknotens. Der über dieses Rohr fließende Gasstrom wird den Agenten sichtbar gemacht als der offizielle Einspeisefluss am jeweiligen Einspeiseknoten. Der Dispatcher-Agent muss also dafür sorgen, dass über die M-Rohre pro Stunde die Richtige Menge an Gas fließt.

Die Rohre der Art G (groß) dienen dazu, Gas zu speichern, damit Änderungen im offiziellen Fluss nur nach und nach Auswirkungen auf den Druck am Einspeiseknoten haben.

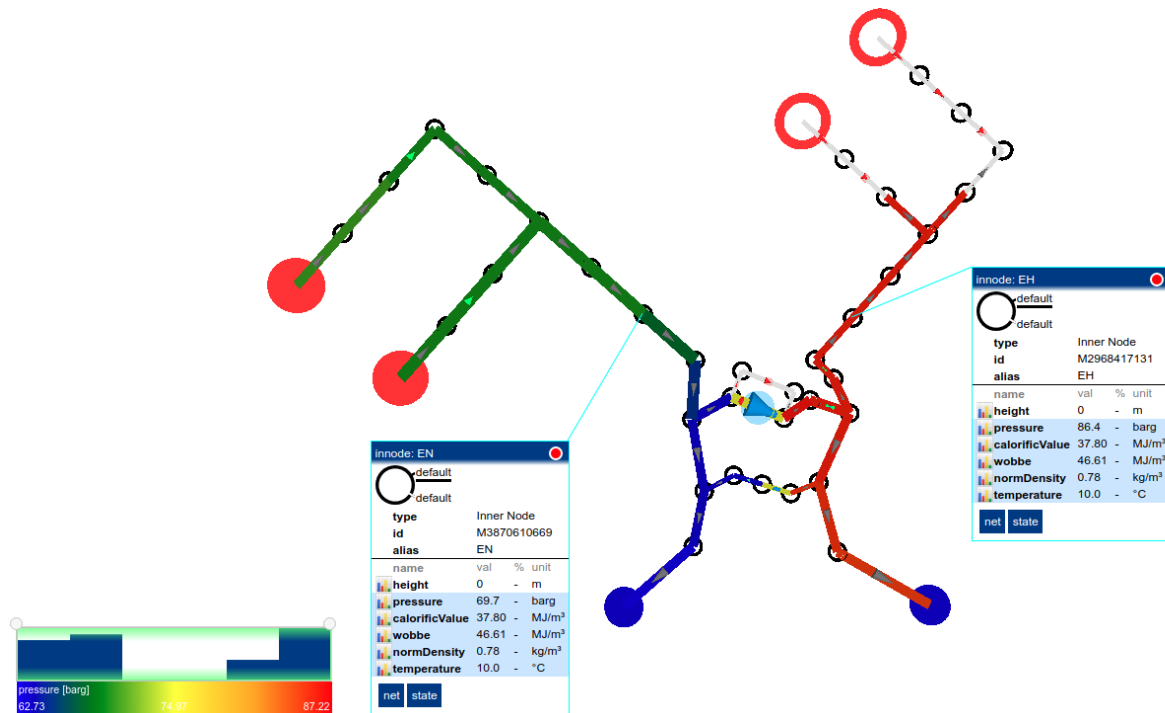
Die Rohre der Art K (klein), dienen dazu, dass durch Fluss im Netz des Einspeiseknotens geeignete Druckabfälle realisiert werden.

Außerdem gibt es noch zwei Rückschlagklappen, die verhindern sollen, dass Gas an den Knoten ND und HD das Netz verlassen kann.

Zusammenfassend kann man sagen, dass die Hilfsnetze dazu dienen das Verhalten von Gasnetzen an den Schnittstellen zu Upstream-Netzen realitätsnah zu simulieren.

Stratopologie

Mit diesem Netz soll gestartet werden. Das Bild zeigt das Originalnetz mit zwei Exits ganz unten und zwei Hilfsnetzen an den Originalentrys EN und EH wie oben beschrieben. Aufgabe des Dispatcher-Agenten ist es im wesentlichen einen Verdichter und einen Regler zwischen einem Hoch- und einem Niederdrucknetzteil zu steuern.



Das Bild kommt aus einem Netzbetrachtungsprogramm, das im Rahmen des Projekts genutzt werden kann.