

Unstrukturierte Berechnung von Verdichterkenfeldern mithilfe von SVM und Antwortflächen

D. Ivanov, BTU Cottbus-Senftenberg, Siemens-Halske-Ring 14, 03046 Cottbus, Deutschland
D. Bestle, BTU Cottbus-Senftenberg, Siemens-Halske-Ring 14, 03046 Cottbus, Deutschland
C. Janke, Rolls-Royce Deutschland Ltd & Co KG, Eschenweg 11, 15827 Blankenfelde-Mahlow, Deutschland

Kurzfassung

Zur Beurteilung der Leistung und Betriebssicherheit von Verdichtern in Turboantriebswerken ist die Kenntnis des Verdichterkenfelds von entscheidender Bedeutung. Dieses fasst signifikante Betriebspunkte zusammen und beschreibt die Betriebsgrenzen durch Verdichterpumpen und -sperrgrenzen. Bei der Analyse der Betriebspunkte mit 3D-CFD führen lange Berechnungszeiten zu einer gesamten Kennfeldberechnungszeit von mehreren Tagen. Um diese Rechenzeit massiv zu reduzieren, bedarf es sowohl neuer Berechnungsstrategien als auch einer Beschleunigung der Strömungsanalyse. Schlüsselstrategien sind zum einen, sich von der üblichen strukturierten Berechnung zu lösen und stattdessen mit unstrukturierten, geeignet bestimmten Stützstellen zu arbeiten, zum anderen eine geschickte Initialisierung der 3D-CFD. Für den unstrukturierten Kennfeldberechnungsprozess werden im Rahmen dieser Arbeit Methoden der statistischen Versuchsplanung in Kombination mit SVM (Support Vector Machine) vorgestellt, welche einen hohen Grad an Parallelisierbarkeit von benötigten CFD-Evaluationen ermöglichen. Die Strömungsinitialisierung erfolgt sowohl durch reduzierte Strömungsmodelle als auch Superposition von bereits bestimmten Strömungslösungen.

Schlüsselwörter: Verdichterkenfeldberechnung, Support Vektor Maschine, Antwortflächen, Computational Fluid Dynamics

1 Einleitung

In der heutigen Triebwerksentwicklung ist die Verwendung komplexer und zeitaufwändiger numerischer Strömungssimulationsverfahren wie 3D-CFD (Computational Fluid Dynamics) unerlässlich. Ein wichtiger Teilbereich, welcher relativ viele zeitintensive 3D-CFD Berechnungen benötigt, ist die Verdichterkenfeldberechnung. Dabei sind zur qualitativen Beurteilung eines Verdichterentwurfs sowohl reguläre Betriebspunkte wie Reiseflug, Start und Landung hinreichend genau abzubilden, als auch die kritischen, den Verdichterarbeitsbereich limitierende Betriebsgrenzen zu detektieren. Die komplizierten Zusammenhänge bei der Bestimmung dieser Grenzen sowie Konvergenzprobleme der 3D-CFD bei ungünstiger Initialisierung von Startlösungen erschweren eine automatisierte Kennfeldberechnung, weshalb man i. Allg. entweder auf 1D-Strömungsberechnungen mit dem Meanline-Code [1] ausweicht oder sich auf eine manuelle Berechnung weniger Betriebspunkte mit 3D-CFD beschränkt. Janke u.a. [2] gelang durch geeignete Maßnahmen die automatisierte Kennfeldberechnung auf Basis von 3D-CFD, jedoch beträgt die Rechenzeit für einen 4,5-stufigen Versuchverdichter ca. 10 Tage, was für eine

industrielle Anwendung noch nicht akzeptabel ist. Zur Beschleunigung der Verdichterkenfeldberechnung wurden daher zwei Grundforderungen formuliert, welche im Rahmen dieser Arbeit umgesetzt werden: Zum einen muss die Anzahl der notwendigen CFD-Berechnungen zur hinreichend genauen Beschreibung des Verdichterkenfelds reduziert werden, wobei diese möglichst parallel auszuführen sind, zum anderen muss die Rechenzeit jeder einzelnen 3D-CFD-Berechnung verkürzt werden. Die erste Forderung wird durch eine Strategie erfüllt, welche sich von der üblichen strukturierten Berechnung, bei der verschiedene Drehzahllinien berechnet und auf diesen jeweils isoliert Pump- und Sperrgrenze gesucht werden, löst und stattdessen mit unstrukturierten Stützstellen arbeitet. Anhand dieser Stützstellen werden zunächst die Pump- und Sperrlinien mit einer auf Support Vector Machine (SVM) beruhenden Strategie bestimmt. Anschließend werden diese Kurven verwendet, um mit zusätzlichen Stützstellen Antwortflächenapproximationen (engl. RSM = response surface method) für die unterschiedlichen Verdichterkenngößen aufzubauen. Die vorgestellte Methode unterscheidet sich von bisherigen Studien zur Anwendung von Me-

thoden des maschinellen Lernens auf die Verdichterkennfeldberechnung mit z.B. neuronalen Netzwerken [3, 4] oder SVM [5]. Dort stehen von Anfang an eine große Menge von Betriebspunkten bereit, während in dieser Arbeit ausgehend von einer sehr geringen Datenmenge der noch unbekannte Betriebsbereich iterativ erschlossen wird.

Jede ausgewertete Stützstelle entspricht einer aufwändigen 3D-CFD-Rechnung, weshalb der zweiten Forderung dadurch Genüge getan werden muss, dass jede einzelne Rechnung besser initialisiert wird, d.h. die Startlösung der CFD-Iteration bereits dichter an der vermuteten Lösung liegt. Dieses kann dadurch geschehen, dass man mit reduzierten Strömungsmodellen initialisiert oder bereits berechnete Strömungsfelder zur Superposition neuer Lösungen verwendet.

Die hier präsentierte Arbeit ist wie folgt aufgebaut: Nach einer kurzen Einführung in die Thematik der Verdichterkennfeldberechnung wird zunächst der Prozess zur Exploration der Verdichterbetriebsgrenzen mithilfe von SVM sowie die Approximation von Verdichterkenngrößen wie z.B. Verdichterdruckverhältnis, Wirkungsgrad und Eintrittsmassenstrom mit Antwortflächen erläutert. Anschließend werden Methoden zur Initialisierung von 3D-CFD mit zweidimensionalen 2D-Euler-Rechnungen und der Superposition von bereits berechneten Strömungslösungen vorgestellt.

2 Grundlagen zu Verdichterkennfeldern

Ein axialer Turboverdichter ist ein offenes System, in dem die vom Triebwerk angesaugte Luft verdichtet und der Brennkammer zugeführt wird. Die Verdichtung findet dabei durch Übertragung von Wellenarbeit der Rotoren auf das Arbeitsfluid statt, wobei hier gesagt werden muss, dass die Druckerhöhung erst durch die jeweils folgenden Statoren geschieht, welche die stark drallbehaftete Strömung axial ausrichten und durch die diffusorartige Passage zwischen zwei Statoren eine Erhöhung des statischen Drucks verursachen. Da ein Rotor-Stator Verbund, auch Stufe genannt, nur eine maximale Druckerhöhung von $p_E/p_I < 2$ erreichen kann, müssen mehrere Stufen in Reihe geschaltet werden, um die üblichen Verdichtungsverhältnisse von 40 – 60 zu erreichen.

Leistung und Betriebsstabilität sind maßgeblich von der Strömung durch die Verdichterpässagen abhängig. Zwei wesentliche Strömungsphänomene limitieren dabei den Betriebsbereich eines Verdichters, das Verdichterpumpen und das Verdichtersperren. Beim Verdichterpumpen kommt es zu einer Strömungsablösung auf mehreren Schaufeln. Als Folge davon kann die Wellenarbeit nicht mehr auf das Fluid übertragen werden und somit ist nicht genügend Energie in der Strömung vorhanden, um dem hohen Gegendruck in der Brennkammer zu widerstehen. Die Strömungsrichtung kehrt sich dann um und die heißen Gase

aus der Brennkammer strömen in den Verdichter zurück, was zu einer Beschädigung bzw. Zerstörung der Schaufeln führen kann. Das Verdichtersperren hingegen tritt auf, wenn die Strömung in den Schaufelpässagen die Machzahl Eins erreicht. Dann kommt es zu einem Verdichtungsstoß, welcher den Durchfluss durch die Pässagen limitiert. Der Triebwerkschub ist dadurch ebenfalls limitiert, denn eine Erhöhung der Kraftstoffeinspritzmenge in die Brennkammer führt dann zu keiner Schuberhöhung.

Die beschriebenen Phänomene schließen ein zusammenhängendes Arbeitsgebiet mit den möglichen Betriebspunkten eines Verdichters ein. Die Darstellung dieser Punkte mit den entsprechenden Kenngrößen wird als Verdichterkennfeld bezeichnet. Wichtige Größen zur Beschreibung eines Verdichterkennfelds sind das Totaldruckverhältnis

$$(1) \quad \pi = \frac{p_{t,E}}{p_{t,I}}$$

zwischen Verdichtereintritt ($p_{t,I}$) und -austritt ($p_{t,E}$) sowie der Wirkungsgrad

$$(2) \quad \eta = \frac{\pi^{\frac{\kappa-1}{\kappa}} - 1}{\frac{T_{t,E}}{T_{t,I}} - 1}.$$

Die Darstellung erfolgt üblicherweise als Funktion von reduziertem Massenstrom am Eintritt

$$(3) \quad \dot{m}_{r,I} = \frac{\dot{m}_I \sqrt{T_{t,I}}}{p_{t,I}}$$

und reduzierter Drehzahl

$$(4) \quad n_r = \frac{n}{\sqrt{T_{t,I}}}.$$

Dabei sind κ der Isentropenexponent, \dot{m}_I der Eintrittsmassenstrom, n die Drehzahl und $T_{t,I}$ die Totaltemperatur am Eintritt. Alternativ ist statt (3) die Verwendung des reduzierten Austrittsmassenstroms

$$(5) \quad \dot{m}_{r,E} = \frac{\dot{m}_E \sqrt{T_{t,E}}}{p_{t,E}}$$

als unabhängige Einflussgröße in 3D-CFD Analysen und in experimentellen Verdichterversuchen üblich.

Die weit verbreitete Darstellung eines Verdichterkennfelds ist eine zweidimensionale Darstellung anhand parametrisierter Linien konstanter Drehzahl, siehe Abb. 1a. Jede dieser Linien ist durch die Pump- und Sperrgrenze begrenzt. Im allgemeinen werden nur einige wenige Drehzahllinien explizit ausgewertet und dazwischen wird anschließend linear interpoliert. Folglich hängt die Genauigkeit des Verdichterkennfelds von der Anzahl der ausgewerteten Drehzahllinien ab.

In dieser Arbeit wird der dreidimensionale Charakter

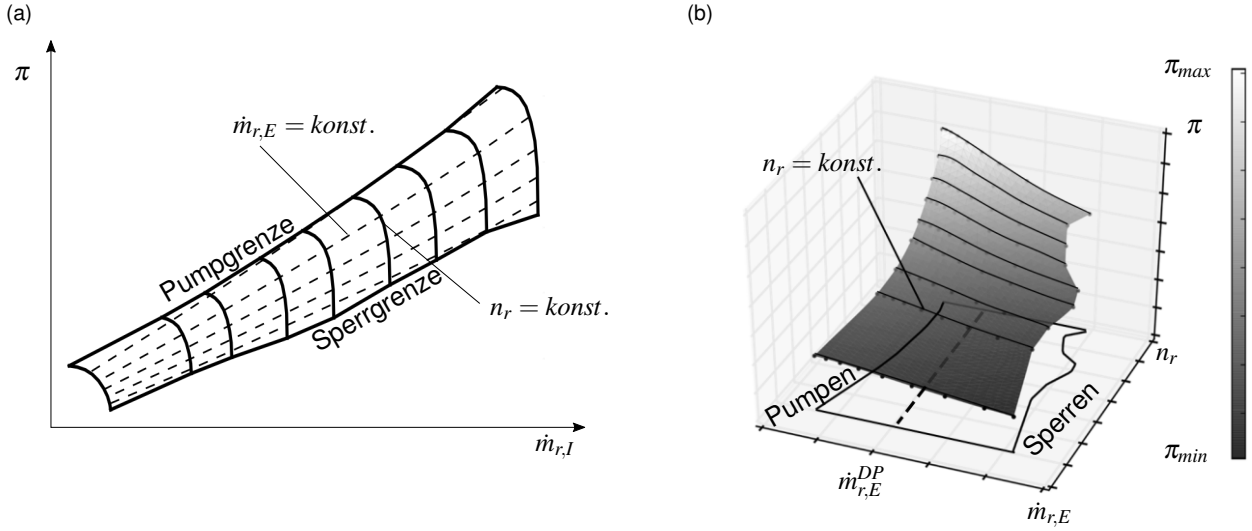


Abbildung 1: Verdichterkennfeld des Totaldruckverhältnisses π in a) klassischer Darstellung mit $\dot{m}_{r,I}$ als unabhängige Variable und n_r als Parameter, sowie b) in dreidimensionaler Darstellung mit π als Funktion von $\dot{m}_{r,E}$ und n_r

des Verdichterkennfeldes erfasst, indem alle Leistungsgrößen als Funktionen des Vektors $\mathbf{x} = [\dot{m}_{r,E}, n_r]^T$ der unabhängigen Variablen (5) und (4) interpretiert werden:

$$(6) \quad \pi = \pi(\mathbf{x}), \quad \eta = \eta(\mathbf{x}), \quad \dot{m}_{r,I} = \dot{m}_{r,I}(\mathbf{x}).$$

Der letzte genannte Zusammenhang ergibt sich aus der Tatsache, dass $\dot{m}_{r,E}$ und $\dot{m}_{r,I}$ sich gegenseitig bedingen und der Zusammenhang durch ein entsprechendes Ersatzmodell hergestellt werden muss.

Die glatte Form der Funktionen (6), wie sie in Abb. 1b beispielhaft für das Totaldruckverhältnis π gezeigt ist, ist prädestiniert für die Verwendung von Antwortflächen. Ein weiterer Vorteil der Darstellung in $[\dot{m}_{r,E}, n_r]$ -Koordinaten ist, dass die Bereiche, welche die Pump- und Sperrgrenzen enthalten, sich strikt durch eine Linie $\dot{m}_{r,E}^{DP} = konst.$ voneinander trennen lassen, siehe Abb. 1b. Dabei ist $\dot{m}_{r,E}^{DP}$ der reduzierte Austrittsmassenstrom für den Auslegungsfall (DP=Designpunkt). Somit ergeben sich die Suchbereiche für die Pumpgrenzlinie zu

$$(7) \quad [\dot{m}_{r,E}^{min}, \dot{m}_{r,E}^{DP}] \times [n_r^{min}, n_r^{max}]$$

und für die Sperrgrenzlinie zu

$$(8) \quad [\dot{m}_{r,E}^{DP}, \dot{m}_{r,E}^{max}] \times [n_r^{min}, n_r^{max}],$$

wobei $\dot{m}_{r,E}^{min}$ und $\dot{m}_{r,E}^{max}$ durch den Benutzer festzulegende Suchgrenzen sind.

Für eine vereinfachte Darstellung werden sich die folgenden Betrachtungen auf den Bereich (7) beschränken. Dazu wird der normierte Vektor

$\mathbf{x} = [x_1, x_2]^T \in [0, 1]^2$ mit

$$(9) \quad x_1 = \frac{\dot{m}_{r,E} - \dot{m}_{r,E}^{min}}{\dot{m}_{r,E}^{DP} - \dot{m}_{r,E}^{min}}, \quad x_2 = \frac{n_r - n_r^{min}}{n_r^{max} - n_r^{min}}$$

eingeführt und alle für die Verdichteranalyse relevanten Größen werden als Funktion von \mathbf{x} dargestellt, d.h.

$$(10) \quad \pi = \pi(\mathbf{x}), \quad \eta = \eta(\mathbf{x}), \quad \dot{m}_{r,I} = \dot{m}_{r,I}(\mathbf{x}).$$

Der im Folgenden entwickelte Prozess ist jedoch einfach auf den Suchbereich (8) übertragbar, indem die unabhängigen Variablen als

$$(11) \quad x_1 = \frac{\dot{m}_{r,E}^{max} - \dot{m}_{r,E}}{\dot{m}_{r,E}^{max} - \dot{m}_{r,E}^{DP}}, \quad x_2 = \frac{n_r - n_r^{min}}{n_r^{max} - n_r^{min}}$$

definiert werden.

3 Auffinden der Verdichterbetriebsgrenzen mit Support Vector Maschine (SVM)

Unter der Annahme, dass eine CFD-Rechnung nur divergiert, wenn Pumpen oder Sperrern auftritt, lässt sich jeder Zustand \mathbf{x}_i durch

$$(12) \quad y_i = y(\mathbf{x}_i) = \begin{cases} +1 & \text{wenn CFD konvergiert,} \\ -1 & \text{sonst} \end{cases}$$

klassifizieren. Für eine Menge aus N analysierten Betriebspunkten (z.B. durch eine DoE = Design of Experiments) ergibt sich dann die Datenbasis

$$(13) \quad \{(\mathbf{x}_i, y_i), i = 1(1)N, \mathbf{x}_i \in [0, 1]^2 \subset \mathbb{R}^2, y_i \in \{-1, 1\}\}.$$

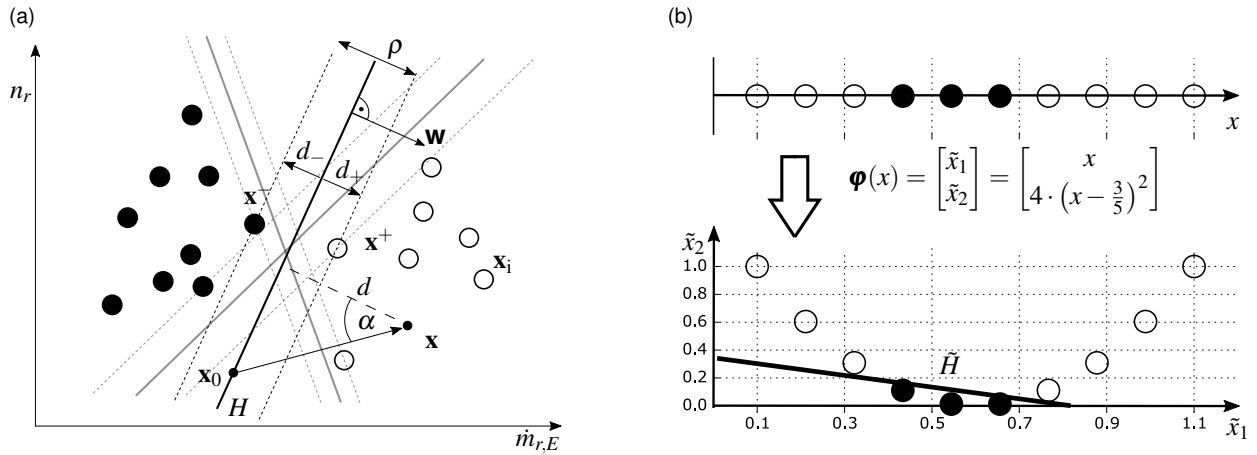


Abbildung 2: Beispiel zur Trennung von gültigen (○) und ungültigen (●) Stichproben durch a) eine lineare Entscheidungsgrenze H und b) einer nicht linear trennbaren eindimensionalen Menge, welche nach einer Transformation ins Zweidimensionale linear trennbar wird.

Die Punkte \mathbf{x}_i beider Klassen lassen sich beispielsweise mit der Methode der Support Vector Machine (SVM) durch eine sog. Entscheidungsgrenze H voneinander separieren.

Nimmt man zunächst an, dass H eine Gerade durch einen Punkt $\mathbf{x}_0 \in H$ mit dem Normalenvektor $\mathbf{w} \perp H$ ist, gilt

$$(14) \quad (\mathbf{x} - \mathbf{x}_0) \perp \mathbf{w} \quad \text{oder} \quad \mathbf{w}^T (\mathbf{x} - \mathbf{x}_0) = 0 \quad \forall \mathbf{x} \in H.$$

Durch Substitution von $b := -\mathbf{w}^T \mathbf{x}_0$ ergibt sich für die Entscheidungsgrenze H die Gleichung

$$(15) \quad H : \mathbf{w}^T \mathbf{x} + b = 0, \quad b \in \mathbb{R},$$

wobei \mathbf{w} und b so gewählt werden müssen, dass z.B.

$$(16) \quad \mathbf{w}^T \mathbf{x}_i + b = \begin{cases} > 0 & \text{für } y_i = +1, \\ < 0 & \text{für } y_i = -1 \end{cases}$$

gilt. Die Entscheidungsfunktion ergibt sich dann als

$$(17) \quad y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) \in \{-1, 1\} \in \mathbf{x} \notin H.$$

Bei Betrachtung aller Trennlinien in Abb. 2a wird ersichtlich, dass mehrere Möglichkeiten zur Anordnung von H existieren. Die Wahl einer optimalen Trennung erfolgt durch die Lösung eines Optimierungsproblems, wobei die Breite des in Abb. 2a mit ρ gekennzeichneten Separationstreifens maximiert wird.

An dieser Stelle wird auf die Herleitung verzichtet und stattdessen auf einschlägige Literaturquellen [6], [7] bzw. die Darstellung [8] der Autoren verwiesen. Als optimale Lösung ergeben sich der Normalenvektor

$$(18) \quad \mathbf{w}^* = -\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i$$

und die Offsetkonstante

$$(19) \quad b^* = -\frac{1}{2} (\mathbf{w}^{*T} \mathbf{x}^+ + \mathbf{w}^{*T} \mathbf{x}^-),$$

wobei die Lagrange-Multiplikatoren λ_i^* Teilergebnisse der Lösung des Optimierungsproblems sind. Die Punkte mit dem kleinsten Abstand zu H , in Abb. 2a durch $\tilde{\mathbf{x}}^+$ und $\tilde{\mathbf{x}}^-$ gekennzeichnet, werden Support Vektoren genannt, da sie die Klassifizierungsgrenze stützen. Werden nun (18) und (19) in (17) eingesetzt, ergibt sich für jedes neue \mathbf{x} folgende Klassifizierung:

$$(20) \quad \begin{aligned} y(\mathbf{x}) &= \text{sign}(\mathbf{w}^{*T} \mathbf{x} + b^*) \\ &= \text{sign} \left(-\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i^T \mathbf{x} - \frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i^T \mathbf{x}^+ - \frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i^T \mathbf{x}^- \right). \end{aligned}$$

Im Allgemeinen sind Entscheidungsgrenzen realer Probleme keine linearen Mannigfaltigkeiten, so dass obige lineare SVM nicht durchgeführt werden kann. Zur Lösung dieses Umstands wird die Kernel-SVM benutzt, welche ein linear nicht trennbares Problem aus dem ursprünglichen Raum mittels einer Funktion

$$(21) \quad \tilde{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}), \quad \boldsymbol{\varphi} : \mathbb{R}^2 \rightarrow \mathbb{R}^D,$$

in einen höherdimensionalen Feature-Raum transformiert, in welchem eine lineare Trennung durch eine Hyperebene \tilde{H} möglich ist. Exemplarisch ist eine solche Transformation einer 1D-Menge $x \in \mathbb{R}$ in den zweidimensionalen Raum $\tilde{\mathbf{x}} \in \mathbb{R}^2$ in Abb. 2b dargestellt. Für einen Punkt $\tilde{\mathbf{x}} \in \tilde{H}$ gilt dann analog zu (15)

$$(22) \quad \tilde{H} : \tilde{\mathbf{w}}^T \tilde{\mathbf{x}} + \tilde{b} = 0,$$

wobei sich Normalenvektor und Offsetkonstante analog zu (18) und (19) als

$$(23) \quad \tilde{\mathbf{w}} = -\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \tilde{\mathbf{x}}_i, \quad \tilde{b} = -\frac{1}{2} (\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}^+ + \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}^-),$$

aus den transformierten Daten $(\mathbf{x}_i, y_i) \rightarrow (\tilde{\mathbf{x}}_i = \boldsymbol{\varphi}(\mathbf{x}_i), y_i)$ ergeben.

Die Transformationsfunktion $\boldsymbol{\varphi}(\mathbf{x})$ und die Dimension D in Gl. (21) sind in der Regel nicht bekannt, weshalb der sog. Kernel-Trick verwendet wird. Bei näherer Betrachtung von Gl. (20) erkennt man, dass nur Skalarprodukte von \mathbf{x} -Vektoren benötigt werden und somit nicht die Transformationsfunktion selbst erforderlich ist, sondern nur das Skalarprodukt davon. Die Kernelfunktion

$$(24) \quad K(\mathbf{x}_i, \mathbf{x}_j) := \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j = \boldsymbol{\varphi}^T(\mathbf{x}_i) \boldsymbol{\varphi}(\mathbf{x}_j).$$

bildet dieses Skalarprodukt ab. Durch geschickte Wahl dieser Funktion und unter Beachtung der Mercer-Bedingungen ist dann eine lineare Trennung im höherdimensionalen Raum möglich. Auch hierbei sei auf die Literatur [6], [7] verwiesen.

Im Rahmen der vorliegenden Untersuchung wird ein Gauss-Kernel der Form

$$(25) \quad K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}\right)$$

mit frei wählbarem σ verwendet. Für eine automatische Wahl wird hier iterativ ein optimales σ bestimmt, das die Breite ρ des Grenzstreifens um \tilde{H} maximiert. Der Wertebereich wird dafür auf $\sigma \in [0.3, 3.0]$ begrenzt, da erfahrungsgemäß höhere Werte zu zackartigen Entscheidungsfrenen führen.

Für einen neuen Betriebspunkt \mathbf{x} lautet die Klassifizierungsfunktion dann analog zu Gl. (20) wie folgt:

$$(26) \quad y(\mathbf{x}) = \text{sign} \left(-\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}) - \frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i (K(\mathbf{x}_i, \mathbf{x}^+) + K(\mathbf{x}_i, \mathbf{x}^-)) \right).$$

Dieser nichtlineare SVM-Ansatz kann nun verwendet werden, um Sperr- und Pumpgrenzen eines Verdichters zu bestimmen. Zunächst wird entsprechend Gl. (9) ein Suchfenster $\mathbf{x} \in [0, 1]^2$ für die Pumpgrenze definiert. Anschließend wird eine Stichprobe aus 10 Betriebspunkten erstellt

$$(27) \quad X = \{\mathbf{x}_i \in [0, 1]^2, \quad i = 1(1)10\},$$

wobei vier die Eckpunkte und einer den Mittelpunkt des Suchfensters repräsentieren. Die übrigen werden

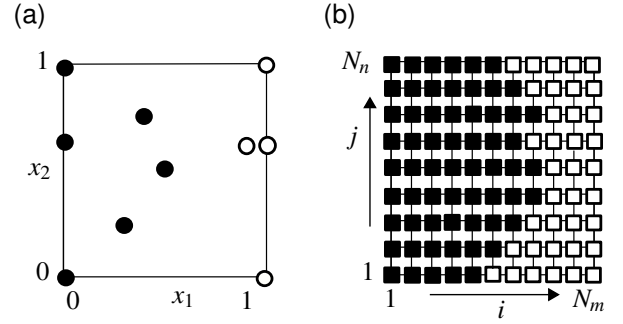


Abbildung 3: Initiale Stichprobenverteilung a) und ausgewertetes Gitter b) mit als gültig (o, □) und ungültig (●, ■) klassifizierten Punkten

zufällig gewählt, siehe Abb. 3a. Nach CFD-Analyse und anschließender Klassifizierung nach Gl. (12) wird die Kernel-SVM durchgeführt. Um die Trennlinie (22) auch im Ausgangsraum $\mathbf{x} \in \mathbb{R}^2$ sichtbar zu machen und damit die Pumpgrenze $x_{p,1} = f(x_{p,2})$ zu approximieren, wird folgender Prozess durchgeführt: Das Analysefenster $[0, 1]^2$ wird mit einem orthogonalen Gitter von $N_m \times N_n$ Punkten

$$(28) \quad \mathbf{x}^{(i,j)} \in [0, 1]^2, \quad i = 1(1)N_m, \quad j = 1(1)N_n,$$

gerastert. Alle Rasterpunkte werden mithilfe von Gl. (26) durch $y^{(i,j)} \in \{-1, 1\}$ klassifiziert, siehe Abb. 3b. Die gesuchte Trennlinie besteht dann aus N_n Punkten, an denen ein Vorzeichenwechsel von $y^{(i,j)}$ passiert:

$$(29) \quad x_{p,2}^{(j)} = x_2^{(k,j)}, \quad f_p(x_{p,2}^{(j)}) = \frac{x_1^{(k,j)} + x_1^{(k+1,j)}}{2}$$

mit $k : y^{(k,j)} y^{(k+1,j)} < 0, \quad j = 1(1)N_n.$

Die so bestimmten Punkte werden anschließend in einer Datenbasis

$$(30) \quad X_p^{(1)} = \left\{ \mathbf{x}_p^{(j)} = [f(x_{p,2}^{(j)}), x_{p,2}^{(j)}]^T, \quad j = 1(1)N_n \right\}$$

gespeichert. Diese erste vorgeschlagene Entscheidungsfrennt entspricht natürlich nicht der echten Betriebsgrenze, da diese auf nur sehr wenigen Punkten basiert. Deshalb ist eine geeignete Erweiterung der Datenbasis notwendig. In [8] und [9] wird diese Datenbasis (27) iterativ erweitert und nach jeder Erweiterung eine SVM Approximation durchgeführt bis $x_{p,1} = f_p(x_{p,2})$ eine fast vollständige Übereinstimmung mit der echten Pumpgrenze erzielt.

Der iterative Algorithmus A zur Erweiterung der Datenbasis ist wie folgt:

- A1 Setze $v := 1$, erstelle initiale Datenbasis X nach Gl. (27), werte CFD-Lösungen für $\mathbf{x}_i \in X$ aus und klassifiziere diese nach Gl. (12).

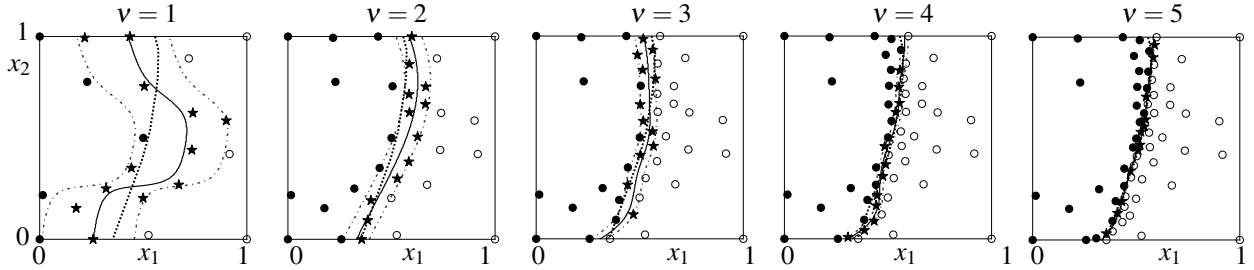


Abbildung 4: Iterative flächenbasierte Exploration der Pumplinie mithilfe von konvergierten (○) und divergierenden (●) CFD-Lösungen sowie Kandidatenpunkten (★) für die Verfeinerung der bestehenden Datenbasis. Als Ergebnis ist die approximierte Pumplinie (durchgestrichene Linie) mit ihren Grenzlinien (strichpunktierte Linie) der echten Pumplinie (gepunktete Linie) gegenübergestellt.

- A2 Bestimme mithilfe der Kernel-SVM die Entscheidungsfunktion (26).
- A3 Wende diese auf ein Raster (28) an und bestimme die Entscheidungsfront $X_P^{(v)}$ entsprechend (29) und (30).
- A4 Für $v < 2$ oder falls nacheinander bestimmte Entscheidungsfronten die Konvergenzbedingung

$$RMS = \sqrt{\frac{1}{N_n} \sum_{j=1}^{N_n} \left(x_{P,1}^{(j)} \Big|_{\in X_P^{(v)}} - x_{P,1}^{(j)} \Big|_{\in X_P^{(v-1)}} \right)^2} \leq \varepsilon$$

nicht erfüllten, bestimme eine Menge C von zusätzlichen Betriebspunkten, klassifiziere diese mit (12) und erweitere die Datenbasis auf $X := X \cup C$. Erhöhe $v := v + 1$ und fahre mit Schritt A2 fort.

Bei der Auswahl der Kandidatenmenge C ist sowohl eine serielle Erweiterung mit einzelnen neuen Kandidaten [8], als auch eine parallele Erweiterung um mehrere Kandidaten [9] möglich. Zur Nutzung des Potenzials eines CPU-Clusters werden in [9] bis zu 12 Betriebspunkte gleichzeitig ausgewertet, wobei nur 6 Iterationen zur hinreichend genauen Approximation der Betriebsgrenzen erforderlich sind. Dieses effiziente Konzept wird deshalb auch in dieser Arbeit verwendet, wobei der in [9] entwickelte Flächenansatz verwendet wird, siehe Abb. 4 für Pumpgrenzlinie. Zur Überprüfung der Genauigkeit der Methode wird diese nun auf ein 1D-CFD Verdichtermodell angewendet. Dabei werden die approximierten Betriebsgrenzen mit den hochgenau evaluierten Werten an neun ausgewählten Stellen verglichen und jeweils die Differenz zwischen realen ($\dot{m}_{r,E}^P, \dot{m}_{r,E}^S$) und approximierten (f_P, f_S) Betriebsgrenzen bestimmt:

$$(31) \quad \begin{aligned} e_{P,j} &= \left| \frac{f_P(n_{r,j}) - \dot{m}_{r,E}^P(n_{r,j})}{\dot{m}_{r,E}^{DP}} \right|, \\ e_{S,j} &= \left| \frac{f_S(n_{r,j}) - \dot{m}_{r,E}^S(n_{r,j})}{\dot{m}_{r,E}^{DP}} \right|, \quad j = 1(1)9. \end{aligned}$$

Aus 50 Wiederholungen der Betriebsgrenzenapproximation werden jeweils mittlere Fehler $\bar{e}_{P,j}$ und $\bar{e}_{S,j}$ gebildet. Diese liegen innerhalb folgender Grenzen:

$$(32) \quad \bar{e}_{S,j} < 2.1\%, \quad \text{und} \quad \bar{e}_{P,j} < 1\% \in j.$$

4 Approximation von Verdichtercharakteristiken mittels Antwortflächen

Ausgangspunkt für die Approximation von Verdichtercharakteristiken sind die approximierte Betriebsgrenzen f_P, f_S sowie alle konvergierten Betriebspunkte, welche zum Auffinden dieser Betriebsgrenzen bereits analysiert wurden, siehe Abb. 5a. Bei genauer Betrachtung sind Gebiete mit einer relativ geringen Punktdichte erkennbar, welche die Qualität der Antwortflächen negativ beeinflussen können. Daher werden zusätzliche Stichproben bestimmt und mit CFD ausgewertet, siehe Abb. 5b. Der Algorithmus zur Bestimmung dieser zusätzlichen N_D Stichproben ist wie folgt:

- B1 Fasse alle konvergierten Datenpunkte aus der Pump- und Sperrliniensuche in der Datenbasis X zusammen, die Kandidatenmenge $C = \{\}$ ist zunächst leer.
- B2 Finde einen Kandidaten mit dem größten Abstand zu allen Punkten in X , welche sich gleichzeitig zwischen $x_1 = f_P(x_2)$ und $x_1 = f_S(x_2)$ befinden:

$$\mathbf{c} = \arg \max_{\mathbf{x} \in S} d(\mathbf{x}, X \cup C) \text{ wobei}$$

$$S = \left\{ \mathbf{x} \in [0, 1]^2 \mid f_P^{(v)}(x_2) \leq x_1 \leq f_S^{(v)}(x_2) \right\}.$$
- B3 Füge \mathbf{c} zu C hinzu, d.h. $C := C \cup \{\mathbf{c}\}$, und wiederhole ab Schritt B2 bis $|C| = N_D$.
- B4 Analysiere alle Punkte $\mathbf{c} \in C$ mit CFD.

Ist eine möglichst homogene Informationsverteilung im Betriebsbereich erreicht, werden Antwortflächenmodelle $\hat{\pi}(\mathbf{x}), \hat{\eta}(\mathbf{x}), \hat{m}_{r,I}(\mathbf{x})$ durch Kombination einer

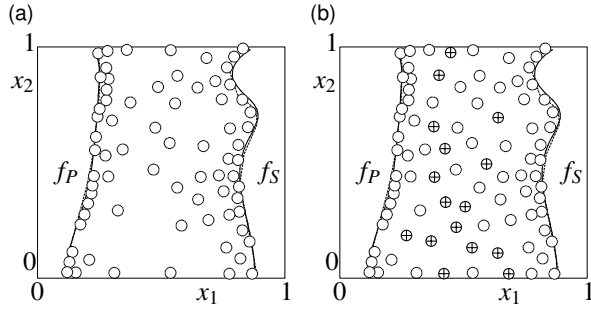


Abbildung 5: Konvergierte Punkte (o) aus a) der Suche nach Betriebsgrenzen und b) mit zusätzlicher fill-up DoE (⊕)

linearen Regressionsfunktion und radialen Basisfunktionen nach [10] erstellt, z.B.

$$(33) \quad \hat{\pi}(\mathbf{x}) = b_0 + [b_1, b_2] \mathbf{x} + \sum_{i=1}^N a_i \psi(\|\mathbf{x} - \mathbf{x}_i\|), \quad N \geq 3,$$

mit $N = |X \cup C|$. Als Kern der radialen Basisfunktionen $\psi(\|\mathbf{x} - \mathbf{x}_i\|)$ wird ein thin-plate-Ansatz

$$(34) \quad \psi(\|\Delta \mathbf{x}\|) = \|\Delta \mathbf{x}\| \ln \|\Delta \mathbf{x}\|$$

verwendet. Die Koeffizienten a_i, b_0, b_1, b_2 werden durch die Interpolationsbedingungen

$$(35) \quad \hat{\pi}(\mathbf{x}_i) = \pi(\mathbf{x}_i) \quad \forall \mathbf{x}_i \in X \cup C$$

sowie die Bedingungen

$$(36) \quad \sum_{i=1}^N a_i = 0, \quad \sum_{i=1}^N a_i \mathbf{x}_i = \mathbf{0}$$

bestimmt.

Die dargestellte Methode zur Approximation wird nun ebenfalls auf ein 1D-CFD Verdichtermodell angewendet. Zur Überprüfung der Genauigkeit der Methode werden für eine Menge \tilde{X} von ausgewählten Betriebspunkten die approximierten Kenngrößen mit tatsächlich berechneten Werten verglichen und die maximalen prozentualen Fehler

$$(37) \quad \begin{aligned} e_\pi &= \max_{\mathbf{x} \in \tilde{X}} \left| \frac{\hat{\pi}(\mathbf{x}) - \pi(\mathbf{x})}{\pi(\mathbf{x}^{DP})} \right|, \\ e_{\dot{m}_l} &= \max_{\mathbf{x} \in \tilde{X}} \left| \frac{\hat{m}_{r,l}(\mathbf{x}) - \dot{m}_{r,l}(\mathbf{x})}{\dot{m}_{r,l}(\mathbf{x}^{DP})} \right|, \\ e_\eta &= \max_{\mathbf{x} \in \tilde{X}} \left| \frac{\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})}{\eta(\mathbf{x}^{DP})} \right| \end{aligned}$$

gebildet. Die vorgestellte Prozesskette wird auch hier 50 mal ausgeführt. Die jeweiligen mittleren Fehler ergeben sich dabei zu

$$(38) \quad \bar{e}_\pi = 2.4\%, \quad \bar{e}_{\dot{m}_{r,l}} = 2.1\%, \quad \bar{e}_\eta = 2.2\%.$$

5 Strategien zur Erstellung des Kennfelds auf Basis von 3D-CFD Rechnungen

Bisher wurden alle Betriebspunkte mit einem sehr schnellen 1D-CFD Löser berechnet. Das angestrebte Ziel ist jedoch die Berechnung des Verdichterkennfelds mit 3D-CFD, wobei hier die hohen Rechenzeiten problematisch sind. Eine möglichst gute Initialisierung der Feldgrößen, d.h. Dichte ρ_i , Druck p_i , Geschwindigkeitsanteile u_i, v_i, w_i sowie Turbulenzgrößen k für jeden Netzknoten kann diesem Problem jedoch begegnen und gleichzeitig die CFD Rechnung robuster machen.

In dem oben beschriebenen Kennfeldberechnungsprozess gibt es drei Szenarios, in den CFD-Analysen auf Basis unterschiedlicher Informationen durchzuführen sind:

Szenario 1: Bei der Auswertung der ersten in Gl. (27) definierten Stichprobe stehen noch keinerlei Strömungslösungen zur Verfügung.

Szenario 2: Bei der iterativen Exploration der Betriebsgrenzen gemäß Algorithmus A stehen bereits die Strömungslösungen der Stichprobe (27) sowie ggf. rekursiv ergänzte Betriebspunkte zur Verfügung. Diese liegen jedoch in unmittelbarer Nähe der approximierten Betriebsgrenzen.

Szenario 3: Bei der fill-up DoE für die Antwortflächen sind bereits die Lösungen aus Szenario 2 bekannt. Die auszuwertende Stichprobe der Kandidatenpunkte liegt innerhalb der Betriebsgrenzen in einem gutartigen Bereich.

Es wird deutlich, dass in den drei Szenarien ein unterschiedlicher Informationsgehalt vorliegt, der für die Initialisierung der Strömungslösung an einem neu zu berechnenden Betriebspunkt genutzt werden kann. Daher werden im Folgenden unterschiedliche Initialisierungsstrategien entwickelt.

In Szenario 1 steht noch keinerlei Information zur Verfügung. Somit ist die einzige Möglichkeit zur Initialisierung von Strömungsfeldern die Verwendung von vereinfachten Strömungsmodellen, die zum einen sehr schnell zu berechnen sind und zum anderen die Tendenz der Hauptströmung hinreichend gut wiedergeben. Hierbei werden zunächst aus den Vorentwurfsdaten über Schaufelgeometrie und allgemeinen Strömungsanforderungen Strömungen auf einzelnen Stromflächen im (r, z, Θ) -Raum berechnet. Dazu werden diese mit einer winkel- und längentreuen Transformation in zweidimensionale (m', Θ) -Stromflächen überführt. Die entstandenen 2D-Schaufelpassagen werden strukturiert vernetzt (Abb. 6) und anschließend wird die Strömungslösung mit dem schnellen 2D-CFD-Löser MISES [11] bestimmt. Dies ist ein erweiterter Euler-Strömungslöser, bei dem die Strömung

im Fernfeld als reibungsfrei betrachtet und die viskosen Effekte in der Grenzschicht mit integralen Grenzschichtgleichungen modelliert werden. Durch Rücktransformation $(m', \Theta) \rightarrow (r, z, \Theta)$ können mehrere dieser 2D-CFD Gitter samt Strömungslösung wieder in das Koordinatensystem des 3D-CFD Volumengitters transformiert werden, siehe Abb. 7. Abschließend werden alle Feldgrößen innerhalb der Schaufelschnitte zur Interpolation auf alle Punkte des 3D-Gitters, und zwischen den gerechneten Stromflächen, genutzt. Zur Überprüfung der Wirksamkeit der Initialisierungsstrategie werden fünf Betriebspunkte $\mathbf{x}_i = (\dot{m}_{r,E,i}, n_{r,i})$ mit den Drehzahlen

$$(39) \quad n_{r,i} \in [65\%, 80\%, 90\%, 100\%, 102.5\%] n_r^{DP}$$

in Bezug zur Arbeitspunktdrehzahl gewählt. Der Austrittsmassenstrom $\dot{m}_{r,E,i}$ wird ausgehend vom Betriebsmassenstrom $\dot{m}_{r,E}^{DP}$ so lange reduziert, bis der 2D-CFD-Strömungslöser keine konvergierte Lösung mehr erzeugen kann. Die letzte konvergierte Lösung wird anschließend für die Initialisierung verwendet, wodurch ein kritischer Betriebspunkt nahe der Pumpgrenze entsteht, wie dies beispielsweise beim Berechnen der Betriebsgrenzen erforderlich ist. Als Vergleich werden dann zwei 3D-CFD-Berechnungen durchgeführt, wobei eine gewöhnlich initialisiert (Initialisierung mit dem Auslegungspunkt) und eine mit der obigen Strategie initialisiert wird. Die Rechnung erfolgt für ein 3D-CFD Modell der letzten zwei Verdichterstufen (vgl. Abb. 7) mit ≈ 3.6 Millionen Netzpunkten mit dem Rolls-Royce Strömungslöser HYDRA [12]. Dabei handelt es sich um einen Reynolds-gemittelten Navier-Stokes (engl. RANS) Strömungslöser. Für diese Untersuchung wird ein Spalart-Allmaras Turbulenzmodell in Kombination mit einem logarithmischen Wandgrenzschichtmodell verwendet.

Die Gesamtrechnenzeiten in Abb. 8 ergeben sich einerseits als Summe aus 2D-Initialisierung und 3D-CFD-Rechnung, andererseits direkt. Die Initialisierung mit den aus MISES bestimmten Lösungen ergibt eine reine 3D-CFD Rechenzeitreduktion von 69%, 67%, 45%, 62% und 40%. Zur Gesamtbeurteilung ist jedoch die Summe aus 2D-CFD, Interpolation und 3D-CFD notwendig, wodurch sich die Gesamtbeschleunigung auf 60%, 57%, 29%, 46% und 25% reduziert, was im Mittel einer Beschleunigung von ca. 43% entspricht.

In Szenario 2 sind bereits einige Strömungslösungen der initialen Stichprobe (27) oder einer erweiterten Datenbasis bekannt, vgl. Abb. 4. Für die Initialisierung neuer Betriebspunkte können davon jedoch nur die konvergierten Strömungsfelder eingesetzt werden. Im Folgenden werden zwei Methoden zur Verwendung solcher Lösungen vorgestellt.

Die erste Methode beinhaltet die Verwendung der nächst liegenden konvergierten Nachbarlösung. Dabei wird für jeden zu berechnenden Punkt der nächs-

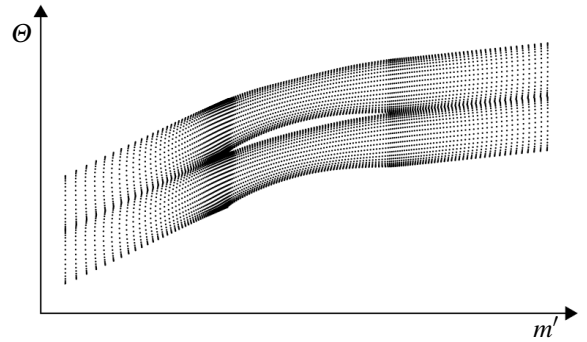


Abbildung 6: Zweidimensionales Rechennetz für eine 2D-CFD-Rechnung in Stromlinienkoordinaten m', Θ

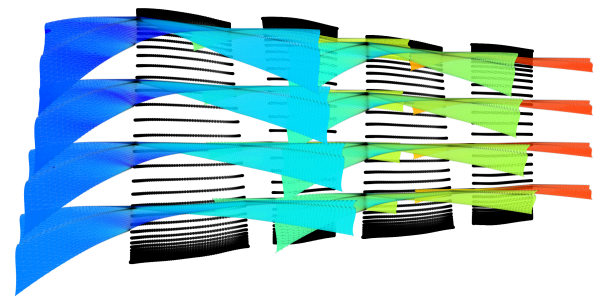


Abbildung 7: Räumliche Anordnung von vier transformierten 2D-CFD-Strömungsfeldern im Koordinatensystem der 3D-Schaukel.

te Nachbar aus der bestehenden Datenbasis von bereits berechneten Punkten bestimmt und dessen Strömungslösung direkt für die Initialisierung verwendet. Bei der Betrachtung des Iterationsverlaufs in Abb. 4 wird deutlich, dass am Anfang die nächsten Nachbarn relativ weit weg von den zu berechnenden Kandidatenpunkten liegen und die Initialisierung daher noch nicht besonders gut sein wird. Mit zunehmender Iterationsanzahl ν rücken die Nachbarn jedoch näher heran, wodurch sich die Rechenzeiten reduzieren (blaue Balken in Abb. 9).

Die zweite Methode basiert auf der Annahme, dass ein Strömungsfeld durch Superposition mehrerer bestehender Lösungen approximiert werden kann. Dazu werden die $N_F = 6$ Strömungsgrößen u, v, w, ρ, p, k jedes dreidimensionalen Strömungsfeldes mit $N_k \times N_l \times N_m$ Netzpunkten unter Einhaltung einer vorher definierten Indizierungsreihenfolge jeweils zu einem Strömungsvektor

$$(40) \quad \Phi_i = \{ (u_1 \dots u_L), (v_1 \dots v_L), (w_1 \dots w_L), (p_1 \dots p_L), (\rho_1 \dots \rho_L), (k_1 \dots k_L) \}_i$$

der Länge $L \cdot N_f$ mit $L = N_k \cdot N_l \cdot N_m$ zusammengefasst. Der Vektor des Strömungsfeldes eines neuen Kandida-

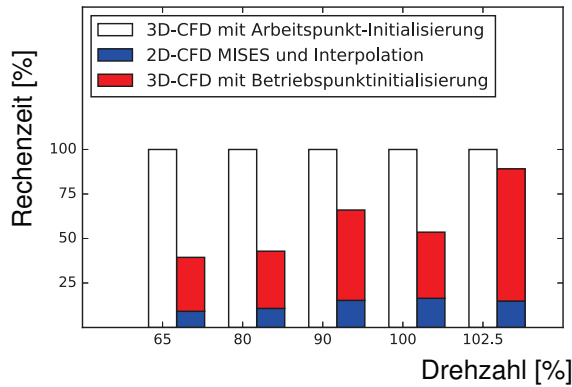


Abbildung 8: Gesamtrechenzeiten der 3D-CFD mit und ohne 2D-CFD Initialisierung.

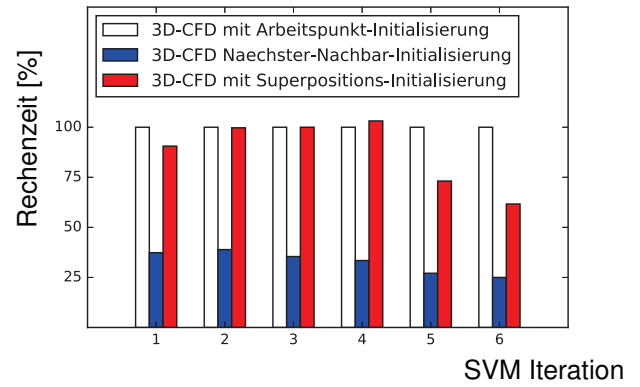


Abbildung 9: Rechenzeitreduktion durch Berücksichtigung bereits bekannter Lösungen im Algorithmus A zur Betriebsgrenzenbestimmung.

tenpunkts \mathbf{x} kann dann mit

$$(41) \quad \Phi(\mathbf{x}) \approx \sum_{i=1}^N \mathbf{a}_i \psi \|\mathbf{x}_i - \mathbf{x}\|$$

approximiert werden, wobei N die Anzahl der bereits bestimmten konvergieren Datenpunkte ist und $\psi \|\mathbf{x}_i - \mathbf{x}\|$ einem thin-plate-spline RBF-Ansatz nach Gl. (34) entspricht. Die Koeffizientenvektoren \mathbf{a}_i ergeben sich aus Interpolationsbedingungen $\Phi(\mathbf{x}) = \Phi_i$ analog zu Gl. (35). Dadurch, dass die Indizierungsreihenfolge vorher festgelegt ist, kann der eindimensionale Vektor anschließend in dreidimensionale Felder zurücktransformiert werden, die der Initialisierung dienen.

Die Verwendung dieser Methode zeigte mehrere Nachteile. Erstes muss für jeden Punkt des Gitters und jede einzelne Feldgröße eine eigene Antwortfläche erstellt werden, was bei diesem Netz mit mit knapp 3.6 Millionen Punkten und 6 Feldgrößen zu durchaus langen Berechnungszeiten führen kann. Der zweite Nachteil ist, dass mit zunehmender Datenbasis die Anzahl N der Basisfunktionen steigt, was ebenfalls zu einer Verlängerung der Rechenzeit führt. Ein weiterer Nachteil ergibt sich bei Betrachtung der Verteilung der Menge von konvergierten Punkten in Abb. 4. Die vorgeschlagenen Punkte liegen größtenteils außerhalb der durch die bereits berechneten Punkte gebildeter Hülle. Somit handelt es sich bei jeder Schätzung um eine Extrapolation, für welche RBF-Ansätze nicht gut geeignet sind. Die Nachteile zeigen sich auch in Abb. 9 in den roten Balken, die kaum eine Ersparnis der 3D-CFD erbringen.

Beim Szenario 3 liegen, ähnlich wie beim Szenario 2, bereits berechnete Betriebspunkte vor. Daher können die für das Vorgängerszenario entwickelten Ansätze auch hier verwendet werden. Bei Betrachtung der fill-up-DoE in Abb. 5b wird deutlich, dass die Verwendung der Methode der nächsten Nachbarlösung

naheliegender ist auch wenn die Kandidatenpunkte nach Algorithmus B so gewählt werden, dass sie einen möglichst großen Abstand zum nächsten, bereits bestimmten Nachbarn haben.

Bei der Verwendung des auf Superposition basieren Verfahrens kann ausgenutzt werden, dass zu jedem vorgeschlagenen Kandidatenpunkt (\oplus in Abb. 5), eine umgebende Hülle aus bekannten Lösungen existiert. Daher entfällt der Nachteil der Extrapolation und neue Strömungsfelder können interpoliert werden.

Zur Überprüfung der Wirksamkeit dieser Initialisierungsmethoden werden die Kandidatenpunkte der fill-up DoE in Abb. 5 mit den bereits berechneten Punkten entsprechend Gl. (41) initialisiert und nachgerechnet. Als reine CFD-Rechenzeitreduktion ergibt sich für die nächste-Nachbar-Initialisierung im Mittel eine Reduktion von 32% und für die Superpositionsmethode im Mittel fast 44%. Bei der Betrachtung der Gesamtbearbeitungszeit einer CFD-Initialisierung und CFD-Berechnung stellt sich heraus das die nächste-Nachbar-Methode fast keinen zusätzlichen Aufwand benötigt und somit die Gesamtrechenzeiterparnis fast unverändert 32% beträgt. Im Gegensatz dazu benötigt die Erstellung der Antwortflächen für viele Stützstellen sehr viel numerischen Aufwand, so dass die Gesamtrechenzeiterparnis gegen Null tendiert.

6 Zusammenfassung und Ausblick

Der vorgestellte unstrukturierte Ansatz zur Berechnung von Verdichterkennfeldern basiert auf Methoden der SVM zur iterativen Exploration der Verdichterbetriebsgrenzen Pumpen und Sperren sowie Antwortflächen zur Approximation aller Verdichterkenngrößen. Der Ansatz erlaubt eine fast beliebige Parallelisierbarkeit von CFD-Analysen und eignet sich somit ideal für den Einsatz auf modernen CPU-Clustern. Auswertungen mit 1D-CFD Verdichteranaysen zeigen einen sehr geringen Approximationsfehler in den entstehen-

den Kenngrößen.

Die industrielle Anwendung dieses Kennfeldberechnungsprozesses bedarf jedoch der Einbindung von 3D-CFD Strömungsanalysen, die derzeit noch immer eine erhebliche Rechenzeit beanspruchen. Daher wurden zusätzlich Methoden zur Beschleunigung von einzelnen 3D-CFD Rechnungen vorgestellt, welche zum einen vereinfachte Strömungsmodelle und zum anderen durch Superposition von bereits bestehenden Lösungen erstellte Strömungsfelder zur Initialisierung nutzen. Durch Verwendung der vereinfachten Strömungsmodelle konnte eine mittlere Zeitersparnis von 43% erreicht werden. Die Initialisierung mittels der Methode des nächsten Nachbarn konnte ebenfalls eine mittlere Beschleunigung von 32% ausweisen. Die Methode der Netzknoteninterpolation konnte durch den enormen Aufwand der Berechnung der Antwortflächen nur geringe Beschleunigungsraten erzielen.

Ein mögliche Alternative zur Strömungsfeldinterpolation ist die Singulärwertzerlegung nach [13] wobei erste Anwendungen auf Verdichterströmungen [14] erfolgversprechende Ergebnisse zeigen.

Kontaktadresse:

ivanov@b-tu.de

Danksagung

Diese Arbeit entstand in Zusammenarbeit mit Rolls-Royce Deutschland im Rahmen des Forschungsprojekts VITIV (Virtuelle interdisziplinäre Triebwerksauslegung mit integrativen Verfahren, Proj.-No. 80164702), finanziert durch das Land Brandenburg und den Europäischen Fonds für regionale Entwicklung. Die Autoren danken Rolls-Royce Deutschland für die Erlaubnis zur Veröffentlichung.

Literatur

- [1] A. Keskin: Process Integration and Automated Multi-Objective Optimization Supporting Aerodynamic Compressor Design, Dissertation, Aachen, Shaker, 2007
- [2] C. Janke, D. Bestle and B. Becker: Compressor Map Computation Based on 3D CFD Analysis, CEAS Aeronautical Journal, vol. 6, pp. 515-527, 2015
- [3] J. Fei, N. Zhao, Y. Shi, Y. Feng and Z. Wang: Compressor Performance Prediction Using a Novel Feed-Forward Neural Network Based on Gaussian Kernel Function, Advances in Mechanical Engineering, vol. 8, 2016
- [4] K. Ghorbanian and M. Gholamrezaei: An Artificial Neural Network Approach to Compressor Performance Prediction, Applied Energy, vol. 86, pp. 1210-1221, 2009
- [5] A. Sodemann, Y. Li, J. Lee and R. Lancaster: Data-Driven Surge Map Modeling for Centrifugal Air Compressors, Proc. of ASME International Mechanical Engineering Congress and Exposition, Chicago, 2006
- [6] V. Vapnik, C. Cortes and L. Saitta: Support-Vector Networks, Machine Learning, vol. 20, pp. 273-297, 1995
- [7] B. Schoelkopf and A.J. Smola: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Cambridge, MA, USA, MIT Press, 2001
- [8] D. Ivanov, D. Bestle and C. Janke: A Response Surface Based Strategy for Accelerated Compressor Map Computation, ECCOMAS EURO-GEN, Madrid, 2017
- [9] D. Ivanov, D. Bestle and C. Janke: Fast Compressor Map Computation by Utilizing Support Vector Machine and Response Surface Approximation, Proc. of. WCCI 2018, Rio de Janeiro, 2018
- [10] A. Bouhamidi and A. Le Méhauté: Radial Basis Functions Under Tension, Journal of Approximation Theory, vol. 127, pp. 135-154, 2004
- [11] M. Drela and H. Youngren: A User's Guide to MSES 2.63, MIT Aerospace Computational Design Laboratory, 2008
- [12] B.L. Lapworth, HYDRA-CFD: A Framework for Collaborative CFD Development, Int. Conf. on Scientific and Engineering Computation, Singapore, 2004
- [13] R. Zimmermann and S. Görtz: Non-Linear POD-based Reduced Order Models for Steady Turbulent Aerodynamics, Proc. of Conference of Applied Aerodynamics, Bristol, 2010
- [14] C. Janke: Ein Beitrag zur automatischen Berechnung von Kennfeldern für Axialverdichter auf Grundlage von dreidimensionaler Strömungssimulation, Dissertation, Aachen, Shaker, 2017
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay: Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011